

ABSTRACT

Title of dissertation: Real Time Terminal Area Trajectory Planning
For Runway Independent Aircraft

Min Xue, Doctor of Philosophy 2006

Dissertation directed by: Assistant Professor Ella M. Atkins
Department of Aerospace Engineering

The increasing demand for commercial air transportation results in delays due to traffic queues that form bottlenecks along final approach and departure corridors. In urban areas, it is often infeasible to build new runways, and regardless of automation upgrades traffic must remain separated to avoid the wakes of previous aircraft. Vertical or short takeoff and landing aircraft as Runway Independent Aircraft (RIA) can increase passenger throughput at major urban airports via the use of vertiports or stub runways. The concept of simultaneous non-interfering (SNI) operations has been proposed to reduce traffic delays by creating approach and departure corridors that do not intersect existing fixed-wing routes. However, SNI trajectories open new routes that may overfly noise-sensitive areas, and RIA may generate more noise than traditional jet aircraft, particularly on approach.

In this dissertation, we develop efficient SNI noise abatement procedures applicable to RIA. First, we introduce a methodology based on modified approximated cell-decomposition and Dijkstra's search algorithm to optimize longitudinal plane

(2-D) RIA trajectories over a cost function that minimizes noise, time, and fuel use. Then, we extend the trajectory optimization model to 3-D with a k -ary tree as the discrete search space. We incorporate geography information system (GIS) data, specifically population, into our objective function, and focus on a practical case study: the design of SNI RIA approach procedures to Baltimore-Washington International airport. Because solutions were represented as trim state sequences, we incorporated smooth transition between segments to enable more realistic cost estimates.

Due to the significant computational complexity, we investigated alternative more efficient optimization techniques applicable to our nonlinear, non-convex, heavily constrained, and discontinuous objective function. Comparing genetic algorithm (GA) and adaptive simulated annealing (ASA) with our original Dijkstra's algorithm, ASA is identified as the most efficient algorithm for terminal area trajectory optimization. The effects of design parameter discretization are analyzed, with results indicating a SNI procedure with 3-4 segments effectively balances simplicity with cost minimization. Finally, pilot control commands were implemented and generated via optimization-base inverse simulation to validate execution of the optimal approach trajectories.

Real-Time Terminal Area Trajectory Planning for Runway Independent Aircraft

by

Min Xue

A Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:

Assistant Professor Ella M. Atkins, Chair/Advisor
Professor Fredric H. Schmitz, Co-Advisor
Associate Professor Robert M. Sanner
Assistant Professor Benjamin Shapiro
Professor Paul M. Schonfeld

© Copyright by

Min Xue

2006

To Shanshan and My Parents.

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Dr. Ella M. Atkins, for her advice, encouragement, and support on research. It was her who revived my interest in the fascinating world of artificial intelligence, helped me to improve my writing and presentation, and provided me opportunities and freedom to enrich my knowledge.

I am also grateful to Dr. Fredric H. Schmitz, for his kind guidance and advice, especially on helicopter acoustics. He is another advisor in my mind. I really appreciate the support of Dr. Ben Wel-C. Sim, who provided me Q-SAM noise database and patient guidance of its use, and also thank Dr. Gaurav Gopalan for his cooperation in preliminary work of my research.

I appreciate the help from Dr. Roberto Celi, who shared his knowledge of helicopter inverse simulation. Special thanks to Kuang Song for his help on usage of geography software package and to Arnat Vale for providing information on BWI airport operations and the flight track data that enabled our development of SNI approach paths.

I would like to thank the other members of my dissertation committee, Dr. Robert M. Sanner, Dr. Benjamin Shapiro, and Dr. Paul M. Schonfeld for their kind support and guidance during the past years.

I would also like to thank all the faculty and staff in Aerospace Engineering, especially in Space System Lab. Many thanks to my dear friends in Alfred Gessow

Rotorcraft Center and Department of Mathematics. It was them who make my life in past years so enjoyable.

My deep gratitude goes to my parents. Without their continuous encouragement and understanding, it would have been impossible for me to accomplish all of my study and research.

Finally, I would like thank my wife, Shanshan Wang. Her love and encouragement is my endless power to keep going ahead.

TABLE OF CONTENTS

Dedication	ii
Acknowledgments	iii
List of Tables	viii
List of Figures	viii
Nomenclature	xii
1 Introduction	1
1.1 Background	1
1.1.1 Runway-Independent Aircraft	3
1.1.2 Simultaneous Non-interfering Procedures	3
1.1.3 Blade-Vortex Interaction Noise	5
1.2 Research Challenges	6
1.3 Problem Statement	7
1.4 Organization of the Dissertation	9
2 Literature Review	11
2.1 RIA Performance and Noise Models	11
2.2 Noise Abatement Procedures	17
2.3 Trajectory Planning Methods	19
2.3.1 Trajectory Optimization	20
2.3.2 Robot Motion Planning	21
2.3.3 Other Related Methods	24
3 Trajectory Planning in the Longitudinal Plane	26
3.1 Trajectory Optimization Algorithm	26
3.1.1 Modified Approximate Cell-Decomposition Method	28
3.1.2 Search/Optimization Strategy	33
3.2 Cost Function	36
3.3 Sensitivity Analysis	37
3.3.1 Noise Only	38
3.3.2 SNI Noise-optimal Approaches	40
3.3.3 Effect of Longitudinal Path Angle Changes	41
3.3.4 Time/Fuel Optimal Solutions	42
3.3.5 Noise and Time/Fuel Trade Study	44
3.3.6 Iterative Deepening Solution Set	46
3.4 Summary	48

4	Terminal Area Trajectory Planning in Three Dimensions	50
4.1	Trajectory Optimization Algorithm	51
4.2	Terminal Area Environment Models	55
4.2.1	Fixed-Wing Traffic Obstacle Model	55
4.2.2	GIS Factor - Population Model	57
4.2.3	Objective Function	59
4.3	Simulation Results	60
4.3.1	Three-Dimensional Optimal Trajectories	60
4.3.2	Effects of GIS Factor - Population	63
4.3.3	SNI Noise-Optimal Trajectories for Varied Entry Sector	65
4.4	Summary	68
5	Trajectory Planning With Smooth Transitions	70
5.1	Q-SAM Sphere Selection for Turning Flight	71
5.2	Smooth Transition Models	73
5.2.1	Longitudinal Transitions	73
5.2.2	Turning Transition	75
5.2.3	Combination of Turning and Longitudinal Transitions	76
5.3	Simulation Results	78
5.3.1	Comparison of 2-D Solutions with Instantaneous and Smooth Transitions	78
5.3.2	Comparison with 3-D Instantaneous Transitions	82
5.3.3	SNI Optimal Solutions with Varied Arrival Sectors	83
5.3.4	DNL Distribution Map	86
5.4	Summary	90
6	Real-time Trajectory Planning	92
6.1	Algorithms	93
6.1.1	Decomposition Strategy	94
6.1.2	Dijkstra's Algorithm	94
6.1.3	Genetic Algorithm	96
6.1.4	Simulated Annealing	96
6.2	Optimization Procedure	99
6.3	Optimization Algorithm Comparison	99
6.4	Analysis of Discretization	103
6.4.1	Effects of Discretization on Design Parameters	103
6.4.2	Effects of Segmented Trajectory	106
6.5	Summary	108
7	SNI Trajectory Simulation and Inverse Simulation	110
7.1	Simulation Model	110
7.2	Optimization-Based Inverse Simulation	111
7.3	Results	114

8	Conclusion	119
8.1	Contributions	122
8.2	Future Work	123
	Bibliography	125

LIST OF TABLES

2.1	Single Segment Approach Cost	19
3.1	Optimal Solution Cost as a Function of η	44
3.2	Noise-optimal Solution Cost with Increasing Depth Level	47
4.1	Comparison With Varying Segment Number in 3D	62
4.2	Comparison of SNI Noise-Optimal Trajectories	68
5.1	Comparison of SNI Noise-Optimal Trajectories with Smooth Transitions	86
6.1	Initial Values for Different Segment Trajectories	101
6.2	Optimal Solutions using Baseline (Dijkstra's) Algorithm	101

LIST OF FIGURES

1.1	FAA Air Traffic Operations Network (OPSNET) Delays	1
1.2	Sequenced Traffic Flow	2
1.3	Runway Independent Aircraft	4
1.4	SNI Operations Architecture [4]	5
2.1	AH-1 Nominal Power Requirement	12
2.2	Effect of <i>miss-distance</i> on BVI noise	13
2.3	Q-SAM BVI Noise Model	14
2.4	Approximate BVI noise characteristics for the AH-1	16
2.5	Single-segment Final Approach Trajectories	18
2.6	Typical Applications of Robot Motion Planning	23
3.1	Cell Neighborhood Illustrating Flight Path Angle Discretization . . .	29

3.2	Example Search Space for Modified Approximated Cell Decomposition	31
3.3	Build Algorithm	32
3.4	Search Algorithm	35
3.5	Iterative Deepening Strategy	36
3.6	Noise Optimal and Worst Approaches from 95 knots to 45 knots . . .	39
3.7	Noise Optimal Approach with Extra Noise Penalty over Sensitive Region	39
3.8	Optimal SNI Approach with A Single Airspace Obstacle	41
3.9	Approach From 95 kts to 45 kts For Varied c_4	42
3.10	Minimum and Maximum Cost Based on Time/Fuel Cost	43
3.11	Optimal Approaches with Varied η	45
3.12	Optimal Trajectory Noise and Time as a Function of η	46
3.13	Noise-optimal Approaches as a Function of Quad-tree Depth	47
4.1	Optimization of Segmented SNI Trajectories.	50
4.2	k -ary Tree Decomposition and Merging (spatial view)	52
4.3	<i>TreeBuild</i> Algorithm	54
4.4	Annual Operations at BWI Airport	55
4.5	East Flow Operation at BWI Airport, July 03, 2003	56
4.6	Obstacles Modeled for East Flow Operation	56
4.7	Population Distribution Map Around BWI Airport	58
4.8	Population Distribution Map (BWI) and Measurement Matrix	59
4.9	3-Dimension 3 Segment Noise-Optimal Trajectory	61
4.10	Effects of Population Density on Noise-Optimal Solution	64
4.11	Path Shape and BVI SEL Distribution for Traffic from Different Sectors	66
5.1	Sphere Selection Module for Turning Flight	71

5.2	Example of Smooth Longitudinal Transition	74
5.3	Example of Smooth Turning Transition	75
5.4	Inclusion of Smooth Pulling in Combined Transition	76
5.5	Inclusion of Smooth Turning in Combined Transition	77
5.6	Solution with 2-D Instantaneous Transitions (depth = 6, $n = \infty$) . .	79
5.7	Solution with Smooth Longitudinal Transition (depth = 6, $n = 0.1$) .	80
5.8	Solution with Smooth Longitudinal Transition (depth = 6, $n = 0.05$)	81
5.9	Solution with Smooth Longitudinal Transition (depth = 6, $n = 0.5$) .	81
5.10	Solution with Smooth Longitudinal Transition (depth = 7, $n = 0.1$) .	82
5.11	3D Optimal Solution with Smooth Transition ($n = 0.05$)	83
5.12	3D Optimal Solution with Smooth Transition ($n = 0.1$)	83
5.13	3D Optimal Solution with Smooth Transition ($n = 0.5$)	84
5.14	Path Shape and BVI SEL Distribution for Traffic from Different Sectors	85
5.15	BVI SEL Difference Map with Varied Arrival Sectors	87
5.16	DNL Distributions with Sector II Approach(100 Flights/Day)	89
5.17	DNL Distributions with Sector II Approach(300 Flights/Day)	89
6.1	Solution Structure	95
6.2	Percentage of Attaining Global Optima	101
6.3	Average Computational Time	102
6.4	Optimal Results for Varied Resolutions	104
6.5	Performance with Varying Resolution Number	105
6.6	Performance with Different Segment Numbers	107
7.1	Trim Setting of Velocities	113
7.2	Matching Trajectory with Overlapping Segments	114

7.3	Time History of Pilot Control Inputs	115
7.4	Difference between Actual and Desired Trajectories	116
7.5	Time History of Pilot Control Inputs	117
8.1	Application of Planning Procedure	121

Nomenclature

\dot{V}	acceleration	ft/s^2
V	flight velocity	<i>knots</i>
g	gravitational constant	ft/s^2
x	ground coordinate with direction from west to east	ft
y	ground coordinate with direction from south to north	ft
z	coordinate upward	ft
α_{TPP}	main rotor tip-path-plane angle	<i>degree</i>
γ	flight path longitudinal angle (in xz plane)	<i>degree</i>
β	flight path lateral angle (in xy plane)	<i>degree</i>
Ω	rotor angular rotation	$1/s$
R	rotor blade radius	ft
μ	advance ratio	$V/\Omega R$
D	rotor drag	lb
W	helicopter weight	lb
SFC	Specific Fuel Consumption	$lb/(hp \times hr)$
C_T	thrust coefficient	1
t	time	s

I_i, K_i, C_i	constant coefficients	
δ_C	collective pitch	<i>degree</i>
δ_A	longitudinal cyclic pitch	<i>degree</i>
δ_B	lateral cyclic pitch	<i>degree</i>
δ_p	rudder pedal	<i>degree</i>

ATC	Air Traffic Control
ATM	Air Traffic Management
VTOL	Vertical TakeOff and Landing vehicle
ESTOL	Extremely Short TakeOff and Landing vehicles
QSRA	Quiet Haul Research Aircraft
RIA	Runway Independent Aircraft
SNI	Simultaneous Non-Interfering
BVI	Blade-Vortex Interaction
GIS	Geography Information Systems
NAP	Noise Abatement Procedure
Q-SAM	Quasi-Static Acoustic Mapping
FATO	Final Approach & TakeOff area
SEL	Sound Exposure Level
dB-PD	decibels Population-Density weighted
DNL	Day-Night average sound Level
GA	Genetic Algorithm
ASA	Adaptive Simulated Annealing

SQP	Sequential Quadratic Programming
NP	Non-deterministic Polynomial time

Chapter 1

Introduction

1.1 Background

The U.S. air transportation system has experienced tremendous growth over the past several decades. Due to the rapidly increasing number of flight operations, the National Airspace System (NAS) has become much more congested, and capacity is one of the primary concerns for both NASA and the FAA. Figure 1.1 shows average monthly delay over the past five years [1]. Although after September 11, 2001, there was a temporary reduction in delay, the NAS has now become even more saturated than before 9/11.

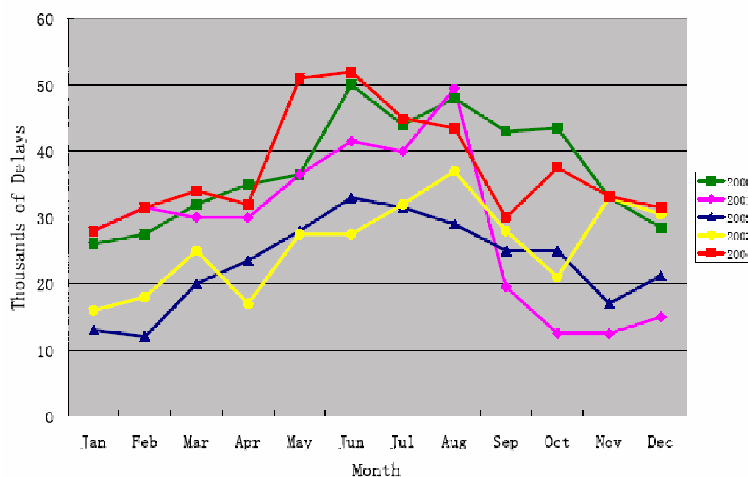


Figure 1.1: FAA Air Traffic Operations Network (OPSNET) Delays

Industry experts predict that flight operations will more than double by 2020. Neither the current air traffic control system nor existing airport infrastructure will

be able to meet the demand. Air transportation delays cost the U.S. \$ 9.4 billion in 2000, and if nothing is improved, the delays are expected to cost \$ 170 billion over the next ten years [2].

Due to traffic queues required for alignment and wake vortex avoidance, sequential air traffic along final approach and initial departure corridors at major urban airports form the primary bottlenecks of the entire air transportation system as illustrated in Figure 1.2. To avoid the wake generated by the previous aircraft, each aircraft is required to maintain a minimum time separation from the previous flight. In urban areas, where congestion is most severe, it is often infeasible to build new runways due to environmental and real estate constraints.

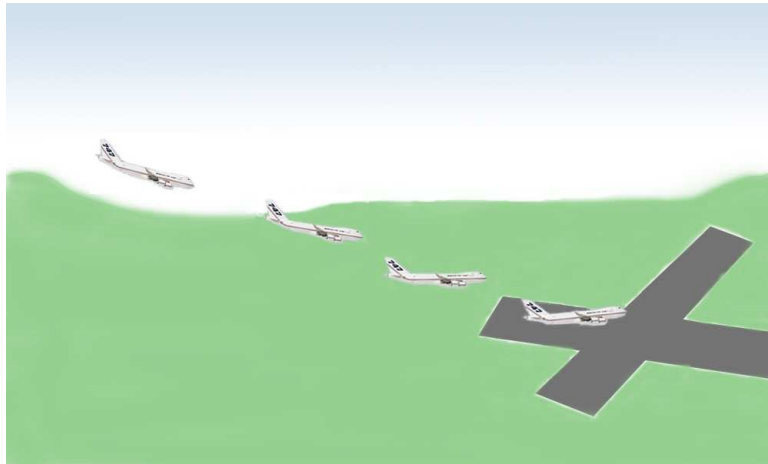


Figure 1.2: Sequenced Traffic Flow

Compared with the existing hundreds of major airports, there are more than 3,400 small airports and numerous urban vertiports that could support additional traffic. The Small Aircraft Transportation Systems (SATS) program [3] jointly sponsored by the FAA and NASA investigated the use of smaller airports to alleviate congestion, but infrastructure and convenience factors limit the ability of suburban

or rural airports to make an appreciable impact.

The goal of this research is to build the models and methods required to effectively integrate runway-independent aircraft traffic into terminal-area airspace without impacting (i.e., reducing the efficiency of) existing traffic queues. This goal is accomplished through the design of simultaneous non-interfering terminal terminal area trajectories that minimize ground noise exposure.

1.1.1 Runway-Independent Aircraft

Runway-Independent Aircraft (RIA) include high-capacity Vertical TakeOff and Landing (VTOL) vehicles, such as the helicopter, tiltrotor (Figure 1.3(a)(b)), as well as powered-lift fixed-wing Extremely Short TakeOff and landing (ESTOL) vehicles, such as the Quiet Haul Research Aircraft (QSRA) (Figure 1.3(c)). The common characteristic of these vehicles is their ability to land/depart via short/stub runways or even heliports. Such vehicles are envisioned to offload short to medium-haul flights (typically less than 400 nm) currently operated with conventional fixed-wing aircraft [4].

1.1.2 Simultaneous Non-interfering Procedures

If the RIA operations are simply added into the existing air traffic system, the workload of air traffic controllers will be increased and delays may become even worse. Thus, the concept of Simultaneous Non-Interfering (SNI) approach and departure procedures has been proposed [4]. For SNI operations, (Figure 1.4),



(a) Helicopter



(b) Tiltrotor



(c) QSRA

Figure 1.3: Runway Independent Aircraft

the new flights are required to not intersect existing fixed-wing routes, so RIA SNI arrivals and departures can be planned independent of fixed-wing traffic, thereby minimizing air traffic control overhead.

The SNI RIA concept is intended to alleviate congestion at the busiest airports. RIA free up landing slots on primary runways by operating on an alternate and potentially existing site such as a short/stub runway or a vertiport. Although traffic management and deconfliction on the ground may be complicated by RIA operations, overall capacity has the potential to increase proportionally to the number of RIA operations.

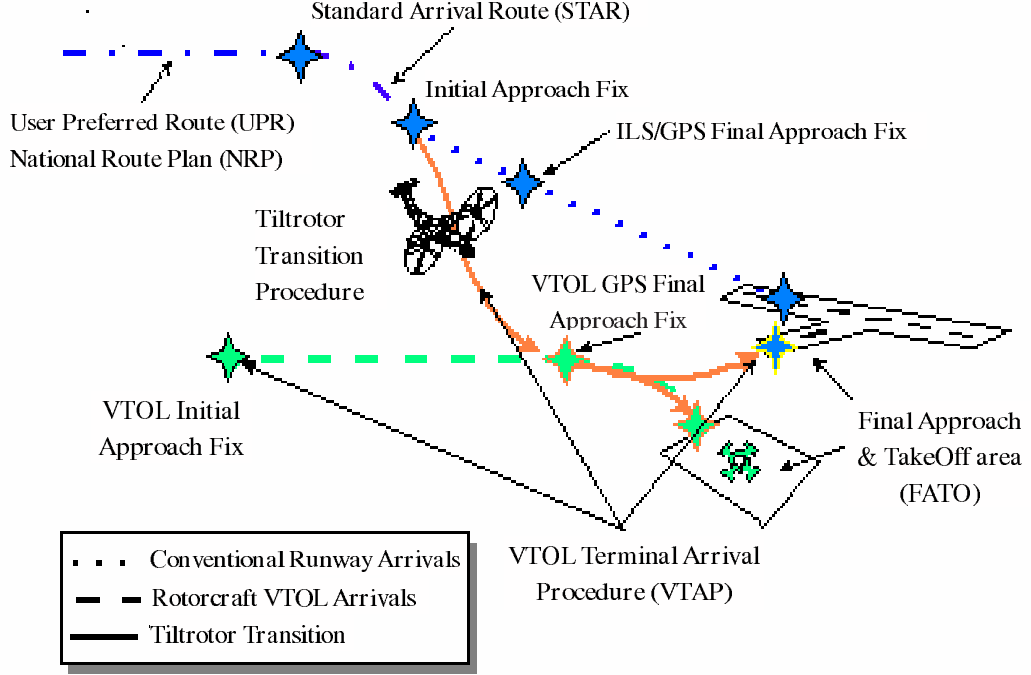


Figure 1.4: SNI Operations Architecture [4]

1.1.3 Blade-Vortex Interaction Noise

RIA (e.g., helicopter, tiltrotor) have the potential to generate much higher noise than fixed-wing aircraft. Due to the existence of experimentally-validated models, we will mainly discuss the noise of a helicopter. Although there are many sources of noise radiated from a conventional single rotor helicopter, when it occurs, helicopter impulsive noise is the most dominant and objectionable [5]. A tiltrotor also experiences this noise during and after conversion to helicopter mode. The characteristic blade “slapping” sound stands out from other noise sources and is often the stated target of community action against helicopter operations. One cause of helicopter impulsive noise arises when the main rotor shed wake operates close to the rotor’s tip-path-plane, causing rapid changes in the local angle of attack on the

main rotor. These rapid changes in the induced velocity flow field at the rotor blades cause impulsive airloads, which in turn, push on the air causing impulsive sources of sound. This noise source has been labeled as Blade-Vortex Interaction (BVI) noise and is the primary cost attribute to be minimized during rotorcraft SNI trajectory optimization.

1.2 Research Challenges

RIA SNI trajectories will occupy previously unused airspace thus may overfly noise-sensitive communities previously undisturbed by fixed-wing traffic. Given the fact that RIAs might generate high noise, as new SNI routes are proposed, public acceptance mandates the development of Noise Abatement Procedures (NAP). To design acceptable RIA SNI procedures the following challenges must be addressed:

- Fixed-wing aircraft noise is dominated by engine noise, which is typically a function of throttle. RIAs have blades or propellers which make the noise function more complex, as illustrated by the BVI phenomenon. Since BVI noise - the typical and dominant noise feature - is non-convex, nonlinear, and experimentally derived, its minimization depends on many parameters. Given the heavily-constrained dynamics, this research aims to solve this NP hard optimization problem within a feasible computational time bound.
- SNI procedures require avoidance of existing fixed-wing traffic corridors. This research investigates the existence of SNI solutions for a crowded airport with consideration of flight envelope limits.

- SNI routes must be designed to minimize the impact of these new routes on populous communities. The non-uniform population distribution near the airport must be incorporated into the optimization process to define routes sensitive to the population under the routes.
- Real-time optimization can provide many benefits. The trajectory planner could be integrated into the cockpit as the part of the flight management system (FMS), and more complex problems like multiple RIA SNI planning or dynamic traffic re-routing based on weather could be accomplished efficiently once the models are integrated. This research investigates a variety of optimization strategies with the goal of identifying SNI trajectories in real-time.
- Finally, given a specific SNI trajectory, the actual trajectory flown must be verified to have approximately the same noise as predicted. Required pilot controls should remain within flight envelope bounds and may be executed autonomously.

1.3 Problem Statement

The major goals of this dissertation are to provide a segmented route design tool that enables identification of acceptable RIA NAPs and to study the design of optimal SNI procedures at a major urban airport. A practical and accurate design procedure is implemented, which computes cost from Geography Information System (GIS) population densities and an experimentally verified BVI noise database, optimizes the trajectory within computational time constraints, and outputs au-

tonomous control commands analogous to those generated by a pilot.

We adopt the AH-1 helicopter as the candidate RIA. For the AH-1, the approach segment is the primary area where main rotor BVI noise is dominant. Thus we study approach procedure design with BVI as the sole noise source. For this noise model, we assume no atmospheric absorption, no wind, no temperature influence, and no ground reflection. Flight plans are typically defined as a sequence of straight-line flight segments with constant path angles and constant accelerations.

To define baseline SNI solutions, we assume the trajectories have instantaneous transitions in both longitudinal and 3-dimensional studies. Next, piece-wise continuous transitions are incorporated to decrease the modeling error during transitions between trim states, and the corresponding trajectory model, thus, is changed to a series of segments linked by smooth push/pull and turning transitions.

To find trajectories well within both the AH-1 operating envelope and passenger comfort region, the longitudinal path angle is constrained by $|\gamma| \leq 9^\circ$, the airspeed is maintained within $38kts \leq V \leq 105kts$, the acceleration is $|\dot{V}| \leq 0.05g$, and, in the three-dimensional airspace case, $|\beta| \leq 80^\circ$ representing a limit of 80° heading change between flight segments.

To illustrate the functionality and applicability of our SNI route planner, we design SNI routes specifically for the Baltimore-Washington International (BWI) airport, a typical congested urban airport. We modeled existing air traffic based on flight track data from July 3, 2003 and modeled the population factor based on 2000 census data downloaded from Tigerline.

Given that these NP-hard trajectory optimizations are actually nonlinear, non-

convex, heavily constrained, discontinuous, and have no gradient-based information, we start from a longitudinal plane (2-D) formulation and developed a *modified approximate cell-decomposition method*, which is combined with Dijkstra’s algorithm to solve the optimal discrete search problem.

After this first solution was evaluated, we focused on a real-world instantiation by modeling population and fixed-wing traffic tracks at a major airport (BWI). We maintained the Dijkstra’s optimization approach but using a k -ary tree as the decomposition method due to its superior ability to model the 3-D search-space.

Although the 3-dimensional problem is solvable by using the optimal Dijkstra’s algorithm approach, the computational time is substantial. We next investigated alternative search techniques with potential efficiency gains - genetic algorithms and adaptive simulation annealing.

Finally, to verify our optimal trajectories could be efficiently tracked with reasonable control input commands, we ran our SNI solutions through a flight simulator [59] and utilized an existing inverse simulation and optimization algorithm [60] to calculate the control state time history.

1.4 Organization of the Dissertation

The organization of this dissertation is as follows:

In Chapter 2, we review AH-1 performance and noise models, and trajectory planning/optimization methods of potential applicability to this work.

In Chapter 3, we present our initial longitudinal plane study. We describe the

modified approximate cell-decomposition algorithm and a variety of 2-D optimal trajectories under many cost function and obstacle scenarios. A sensitivity analysis is presented to study tradeoffs among noise, time, and fuel in the SNI solutions.

Chapter 4 focuses on the real-world models and extension of the optimization to 3-D. We model the GIS factor and fixed-wing traffic tracks based on available statistics, and incorporated the Quasi-Static Acoustic Mapping (Q-SAM) BVI noise model [42]. We generate the 3-D noise-optimal trajectories and analyze the effect of population distribution for a series of SNI NAPs into BWI airport.

In Chapter 5, we model the smooth transitions between trajectory segments to make the trajectory model more realistic and accurate. The effect of such inclusion is analyzed, and a Day-Night average sound Level (DNL) noise map is presented.

In Chapter 6, we study the application of alternative optimization algorithms, specifically the genetic algorithm (GA) and adaptive simulated annealing (ASA), to our trajectory design problem. Additionally, we assess the effects of search-space discretization and the tradeoff between solution accuracy (i.e., cost) and time complexity.

In Chapter 7, we utilized an existing quasi-static 6-DOF model and optimization-based inverse simulation to characterize performance in simulated flight. We generated a realistic series of pilot control commands from optimal segmented trajectories.

Chapter 8 concludes this thesis and discusses future work.

Chapter 2

Literature Review

2.1 RIA Performance and Noise Models

RIA are still in the conceptual stage with respect to both class (e.g., rotorcraft, tilt-rotor, ESTOL) and specific vehicle design characteristics. The SNI airspace model and optimization methods presented in this work are general for any aircraft type. However, presented results are based on an AH-1 rotorcraft, enabling the use of experimentally validated performance and noise models for all optimization process.

The optimization cost for our initial longitudinal study includes time, fuel, and noise terms to enable efficient and quiet procedure design. Time is directly computed from the flight trajectory, while fuel used up to time t_i is derived from a standard rotorcraft model [41]:

$$m_{fuel,i} = \sum_{k=1}^i SFC \times HP_k \times t_k \quad (2.1)$$

where SFC is Specific Fuel Consumption, HP_k is power required per hour for trim flight segment k , and t_k is the corresponding time. HP_k can be expressed as:

$$HP_k = \rho A (\Omega R)^3 \left[\frac{k C_T^2}{2\mu} - \mu \alpha_{TPP} C_T + \frac{s C_{d0}}{8(1 + 4.6\mu^2)} \right] \quad (2.2)$$

where $\alpha_{TPP} = -D/W - \gamma - \frac{\dot{V}}{g}$, D is a function of $(V - f(V^2))$, and $\mu = V/(\Omega R)$. $\rho A(\Omega R)^3$, C_T , s , C_{d0} , and k can all be treated as rotorcraft model-specific constants during approach.

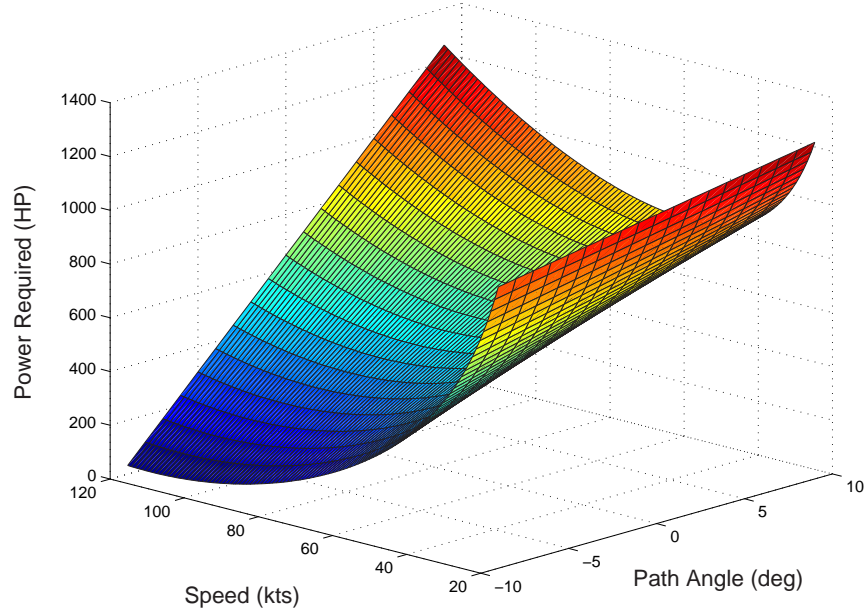


Figure 2.1: AH-1 Nominal Power Requirement

Although engine power should include tail rotor power and installation losses, these are secondary effects and may be considered independent of flight condition in the longitudinal plane. Thus, given Eqns. 2.1 and 2.2, fuel consumption is a function of V and γ . For the AH-1 helicopter, the nominal power requirement is shown in Figure 2.1.

Many efforts have been devoted to research of BVI noise characteristics. A sketch of a side view of a helicopter with its shed wake is shown in Figure 2.2(a) for a helicopter in normal level flight [6]. BVI noise is not seen as a problem in this nominal cruising condition because of high *miss-distance* between the previously

generated wake and blades. However, when the helicopter descends or decelerates in preparation for a landing, the rotor's shed wake can be forced to operate close to the rotor blades, causing strong BVI noise (see Figure 2.2(b)). Further increases in descent/deceleration may cause the rotor wake to operate above the tip-path-plane of the helicopter (see Figure 2.2(c)), causing reductions from the peak BVI noise level [43] [42].

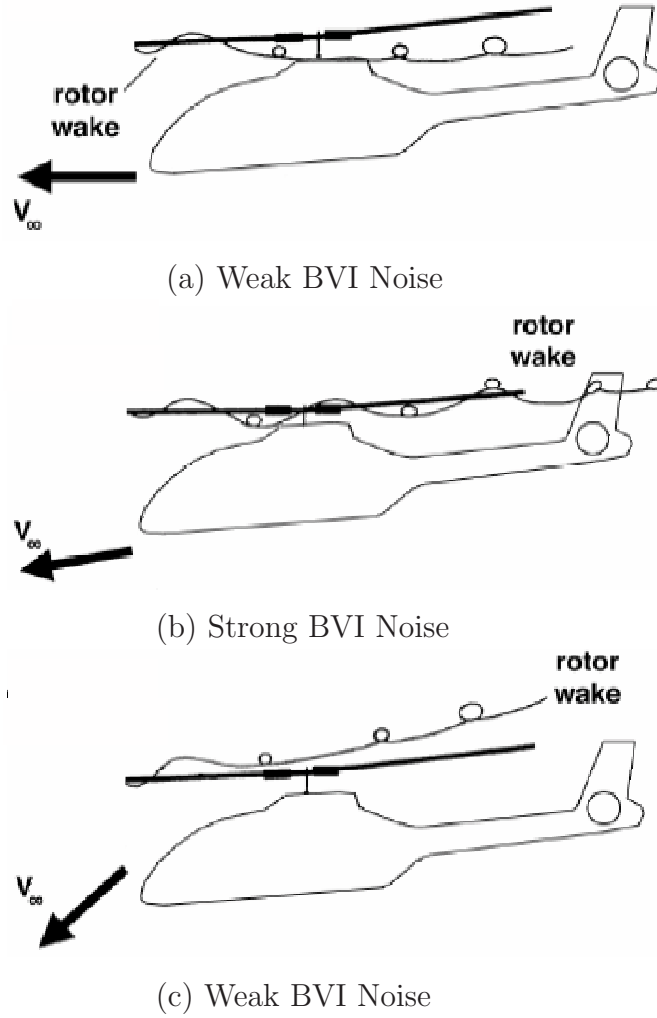


Figure 2.2: Effect of *miss-distance* on BVI noise

An empirical BVI noise database, Q-SAM [42], has been developed to predict

the magnitude of BVI noise and conceptually depicted in Figure 2.3. The output is the A-weighted Sound Exposure Level (SEL) expressed in decibels ($dB A$).

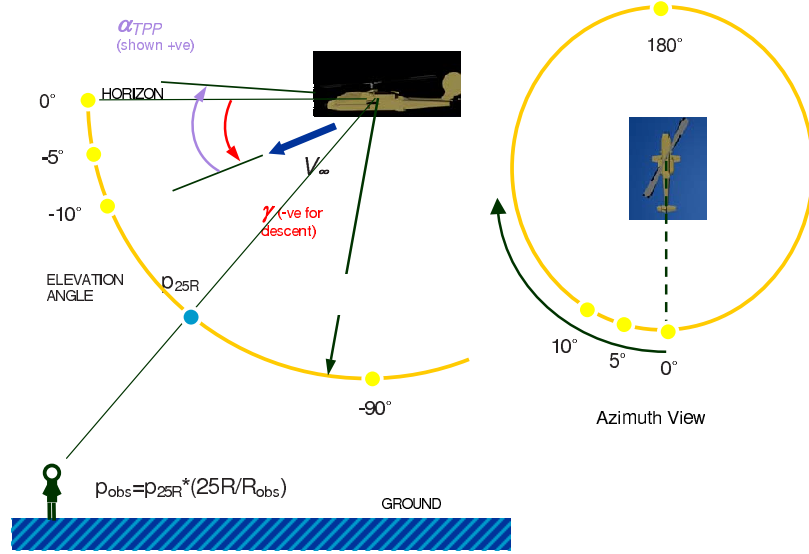


Figure 2.3: Q-SAM BVI Noise Model

Here R_{obs} is the distance between the observer and the vehicle, elevation angle θ is counter-clockwise and starts from the horizon, and azimuth angle ψ is clockwise with zero angle pointing back from the rotorcraft. Spherical distributions of radiated noise are developed as a function of tip-path plane angle α_{TPP} and advance ratio μ . Interpolation is performed when the observer is not located at the exact grid point of the sphere defined in the noise database. In the quasi-static model, rotorcraft trim state transitions (e.g. longitudinal path angle, lateral path angle) are presumed instantaneous, and again, we do not include effects from atmospheric absorption, wind, temperature, or ground reflection. Given a segment (two extreme ends i and $i - 1$) with constant γ and \dot{V} , for an observer on the ground, the BVI noise would be:

$$P_{i,i-1} = q(\alpha_{TPP_{i,i-1}}, \mu_{i,i-1}, R_{obs_{i,i-1}}, \theta_{i,i-1}, \psi_{i,i-1}) \quad (2.3)$$

where P refers to the average A-weighted Sound Exposure Level SEL_{av} expressed in decibels (dB A), $R_{obs_{i,i-1}}$ is the average observer distance. The approximate trends of BVI noise are similar to Figure 2.4 in this chapter.

Considering the computational cost of direct lookup and interpolation of the Q-SAM database, for our initial longitudinal study, Gopalan [6] derived the following analytical longitudinal-plane noise function Eqn. 2.4 based on curve-fitting:

$$P(K_i, C_i, I_i, V_i, x, z) = K_1(1 + \mu)^{K_2^2} + C_1 \times 10 \log_{10} \left(\log_{10} \left(1 + \frac{C_2 A_0 (z_1/z)^5}{P} \right) \right) + C_3 \begin{cases} -20 \log_{10} (1 + I_1 \mu^2 (\alpha_{TPP,0} - \alpha_{TPP})^2) & \text{if } \alpha_{TPP} < \alpha_{TPP,0} \\ -20 \log_{10} (1 + I_2 \mu^2 (\alpha_{TPP,0} - \alpha_{TPP})^2) & \text{if } \alpha_{TPP} > \alpha_{TPP,0} \end{cases} \quad (2.4)$$

In Eqn. 2.4, P refers to the average SEL_{av} expressed in decibels (dB). I_1 is a function of the advance ratio μ for a specific BVI, and the C_i and K_i are constants computed from a curve-fit of experimental trends. $\alpha_{TPP,0}$ is the rotor tip-path-plane angle corresponding to wake vortex zero miss distance and is a function of V . This empirical noise model represents the average sound power (SEL_{av}), due to BVI, radiated by the helicopter over a representative observer plane a distance z below the helicopter [6]. P will be integrated over the flight trajectory from time t_0 to t_1 to provide a single noise term in the objective function used for this work:

$$f_{noise} = \int_{t_0}^{t_1} 10^{P/10} dt \quad (2.5)$$

To gain insight into BVI behavior as a function of velocity, acceleration, and flight path angle (Eqn. 2.4), Figure 2.4 illustrates typical BVI noise (P) for the AH-1 rotorcraft at four flight speeds ranging from 45 to 105 knots. Note the dependence of noise on flight path angle and the movement of a central ridge defining the peak noise region as velocity V is varied. This velocity dependence complicates the identification of acceptable accelerating/decelerating SNI NAPs as will be demonstrated later.

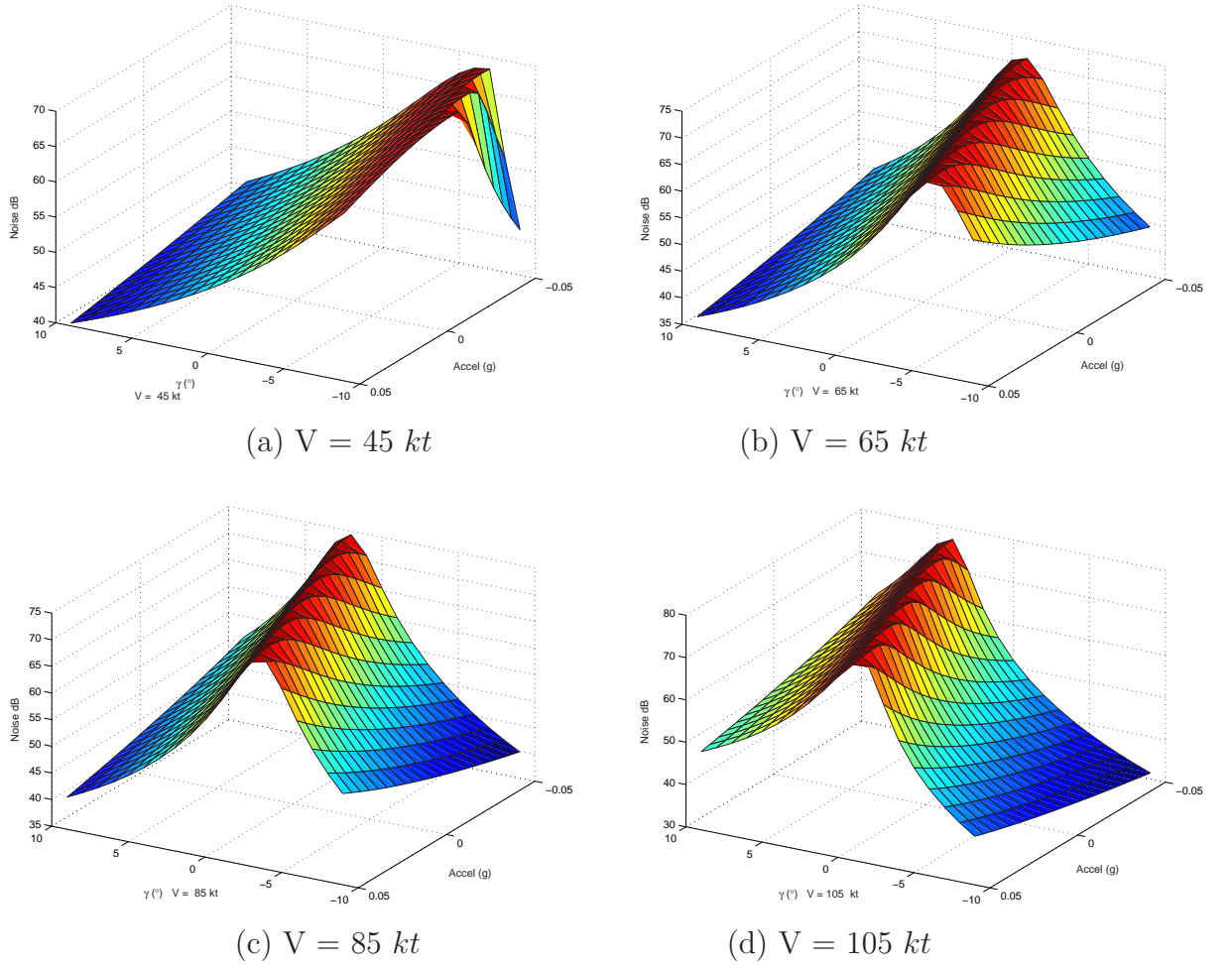


Figure 2.4: Approximate BVI noise characteristics for the AH-1

2.2 Noise Abatement Procedures

Noise abatement procedure (NAP) definition for fixed-wing aircraft has received significant attention. Studies have determined that pilots prefer to fly the simplest single-segment trajectories available [7], resulting in the adoption of *Continuous Descent* approaches [8] that simultaneously lower noise and fuel use due to reduced engine thrust. Researchers have also proposed the optimization of approach [9] and departure [10] trajectories over fuel and noise cast as number of awakenings, computed from a model of radiated ground noise and population density. In such work, numerical optimization techniques identify a continuous (or piecewise continuous) optimal trajectory, enabling accurate computation of objective function quantities such as noise, fuel, and time. As an alternative approach, segmented routes [11] have been designed to facilitate comprehension by pilots and ATC and to enable specification as *procedures* executable with and without advanced automation.

As mentioned previously, fixed-wing aircraft noise is dominated by engine noise, which depends on throttle and can be described analytically. Because the optimization of fixed-wing procedure is therefore much simpler than for RIA, the above work was focused on the study and analysis of NAP impact on the community and pilot workload. Since our work needs to construct SNI routes for rotorcraft, we introduce a more complex noise model (BVI) and optimize routes around obstacles that represent occupied airspace.

To provide a baseline comparison between typical fixed-wing and potential RIA NAPs, Figure 2.5 shows a set of single-segment approach trajectories with varied γ ,

V , and \dot{V} values. Case I-III depict three different velocity profiles along a standard 3° descent approach ($\gamma = -3^\circ$), while Cases IV and V-VII utilize the same set of velocity profiles along 6° and 9° descent paths, respectively. Each approach begins at a lateral distance of 40000 ft (6.6nm) from the Final Approach and TakeOff (FATO) area. Table 2.1 summarizes the noise (Eqn 2.5), time, and fuel(Eqn. 2.1) requirements for each candidate NAP. Consider the 3° standard descent approach. As shown in Figure 2.4, significant BVI noise occurs at mid-range velocities ($60 - 90\text{ knots}$) for shallow descent paths, resulting in high, integrated noise levels for Case I ($V = 70$) and Case III (decelerating). Although Case II ($V = 45$) radiates less noise, efficiency is poor, with over twice the fuel required to maintain the prolonged shallow descent.

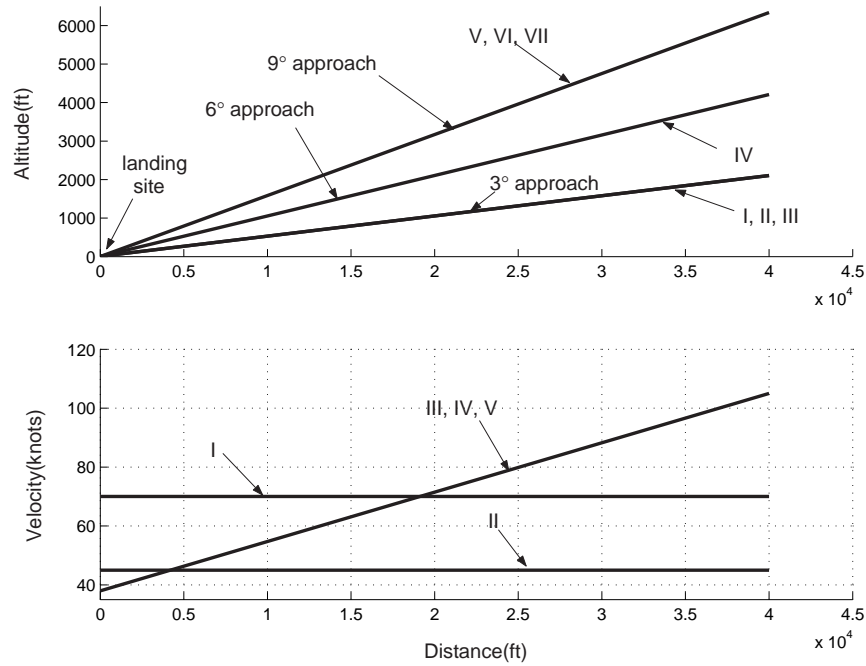


Figure 2.5: Single-segment Final Approach Trajectories

The remaining cases illustrate the effects of increased descent path angle. The

Table 2.1: Single Segment Approach Cost

Case	γ	$V(kt), \dot{V}(ft^2/s)$	BVI Noise (SEL_{av} , dB)	Time(s)	Fuel(lb)
I	-3°	$V = 70, \dot{V} = 0$	89.4	338.8	21.98
II	-3°	$V = 45, \dot{V} = 0$	81.8	527.1	46.85
III	-3°	$V_i = 105, V_f = 45, \dot{V} = -0.01g$	92.8	317.6	21.45
IV	-6°	$V_i = 105, V_f = 45, \dot{V} = -0.01g$	90.0	318.9	14.99
V	-9°	$V = 70, \dot{V} = 0$	77.0	342.6	9.03
VI	-9°	$V = 45, \dot{V} = 0$	89.6	532.9	34.2
VII	-9°	$V_i = 105, V_f = 45, \dot{V} = -0.01g$	86.1	321.1	8.49

6° decelerating descent (Case IV) shows only minor noise reduction relative to the 3° case, with some additional improvement observed for the decelerating 9° descent (Case VII). However, the higher velocity Case V ($V = 70$) now radiates less noise than Case VI ($V = 45$) since peak BVI noise for steep descent paths occurs at low velocities (Figure 2.4). Overall, Table 2.1 suggests Case V, a 9° constant high-velocity descent path and moderate flight velocity. Two factors must be addressed before this solution can be implemented, however. First, the 70-knot velocity must be reduced prior to landing. Deceleration magnitude is limited by passenger comfort and safety considerations - pilots will not prefer to arrive at the FATO area in a steep descent with extra energy. Second, the single-segment solution is only SNI if no fixed-wing airspace passes through this segment.

2.3 Trajectory Planning Methods

Trajectory planning has a rich heritage in multiple research communities. We divide available methods into two groups: *trajectory optimization methods*, the tra-

ditional methods from engineering, and *robot motion planning*, developed somewhat independently in robotics and computer science. Other specialized methods have been used efficiently for certain problems.

2.3.1 Trajectory Optimization

Trajectory optimization has been involved in many engineering applications. Typical Aerospace application requires trajectories that minimize fuel [12], time, radar exposure [13], or propagated noise [9, 10]. For robotics applications, cost functions include time and path length, and obstacle avoidance is required [14, 15, 16].

Typically, system dynamics are defined by a set of ordinary differential equations. Two fundamental methods are available for trajectory optimization problems [17]: the “indirect methods”, which use the calculus of variations or the Maximum Principle of Pontryagin, and the “direct methods”, which transform the original optimal control problem into a nonlinear parameter optimization problem by using multiple shooting methods and direct collocation methods. There are softwares packages to implement these strategies, like Collocation and Multiple-shooting Trajectory Optimization Software (CAMTOS), which uses a combination of direct multiple shooting and direct collocation methods [18].

Both “indirect methods” and “direct methods” need gradient-based information. For instance, after transferring trajectory optimization problem to a typical nonlinear parameter optimization by using “direct methods”, most methods rely on

a Sequential Quadratic Programming (SQP) solver, which needs gradient information. Our main cost terms, noise and population density, have been experimentally determined, thus have no gradient information. We also have complex airspace obstacles, so only dynamic programming is a candidate from the above methods.

For research in air traffic management, in [32], the authors normally solve an en-route conflict identification problem. They utilized the level set method by modeling the problem as hybrid systems with Hamilton-Jacobi-Bellman (HJB) equations. The situations are quite different from our case. First, the focus was on the development of a control strategy for hybrid systems, which combines discrete systems and continuous systems. Second, the cost is only related to distance, a function that could be represented analytically. Finally, the en-route problem defined in [32] is represented as 2 dimensional airspace (lateral plane).

2.3.2 Robot Motion Planning

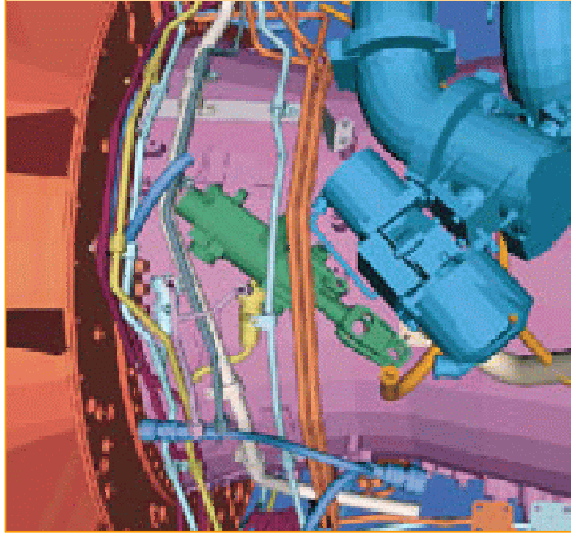
Beyond this scope of “standard” trajectory optimization methods, there exist “path planning” methods developed by the robotics community. The methods include *roadmap*, *potential field*, and *cell decomposition* as described by Latombe [19].

Roadmap methods capture the connectivity of free space in a network of one-dimensional curves, called the *roadmap*, which are used as a set of standardized paths. This method maximizes robot clearance from obstacles but may not be optimal. The *potential field* method sets up an “attractive potential” and a “repulsive potential” around the goal and obstacles, respectively. The negated gradient of the

total potential is treated as an artificial force applied to the robot. At every stage, the direction of this force is considered the most promising direction of motion. This method is neither optimal nor complete. There is nontrivial probability of becoming stuck in a local minimum of the potential. *Cell decomposition* methods decompose the workspace into grids which are either occupied or free. If a path between the initial configuration and the goal configuration exists, the solution could be computed through the midpoints of the intersections of every two successive cells.

Probabilistic roadmap (PRM) [20] and *Rapidly-exploring Random Tree* (RRT) [21] are currently popular motion/path planning techniques as well. *Probabilistic roadmap* was originally developed for holonomic robots in a static environment. The core operations of basic PRM consist two phases, a building phase and a query phase. In the building phase, points are generated at random from a probability distribution over the configuration space of the robot, and a local planner is a computationally inexpensive heuristic for determining if there is a path between two given points. Based on the generated random roadmap, the query phase finds a path between initial and goal configurations by connecting them to the roadmap and searching the roadmap for a sequence of edges linking them. *Rapidly-exploring Random Tree* is a data structure and algorithm suite designed for searching nonconvex high-dimensional spaces. It is constructed incrementally in a way that quickly reduces the expected distance of a randomly-chosen point to the tree, and is particularly well-suited for path planning problems that involve obstacles and differential constraints (nonholonomic or kinodynamic).

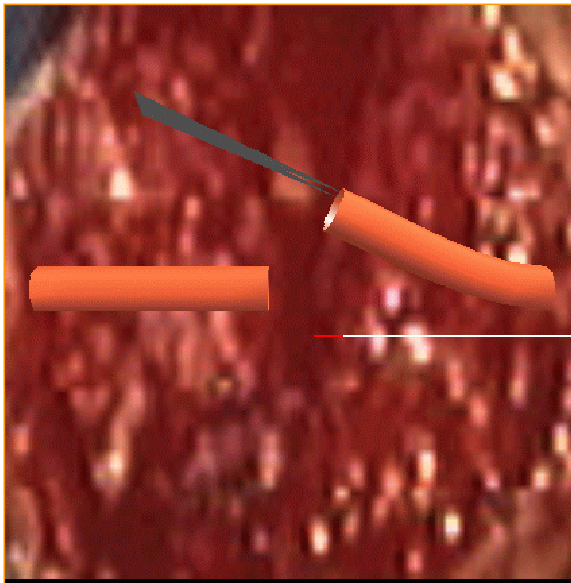
Typical motion planning applications illustrated in Figure 2.6 include manip-



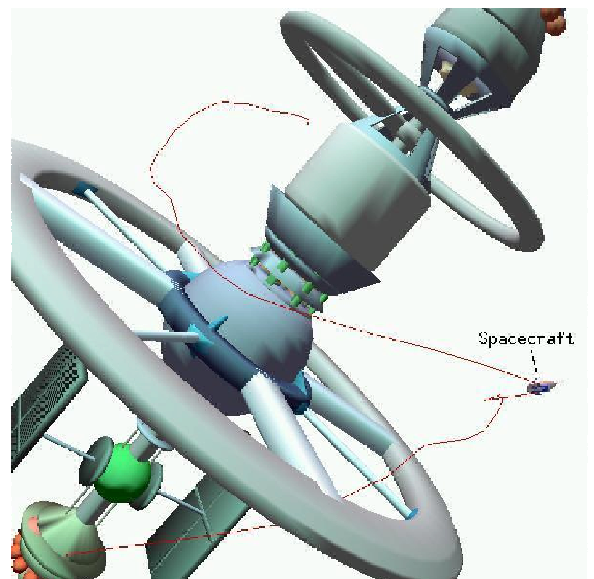
(a) Assembling Planning



(b) Digital Actor Motion Planning



(c) Computer-Aided Surgery



(d) Spacecraft Path Planning

Figure 2.6: Typical Applications of Robot Motion Planning

ulation planning [22], assembly planning [23], and mobility planning [24]. Path planners are also used for computer-aided surgery [25], treatment planning [26], or even for cartoon animations [27]. Planning applications with dynamics constraints have also emerged [28].

Normally, path planning methods are reactive, quickly identifying a solution with only a few dynamics constraints, and they could achieve sub-optimal solutions even for simple costs, usually shortest length paths. The advantage is that most path planners operate without gradient information. Although path planners are challenged by dynamics constraints and velocity/acceleration-dependent cost (e.g. fuel use), they are still promising methods to adapt for our optimization problem. As an example, the roadmap method using Voronoi diagrams has been extended for use during aircraft/UAV trajectory optimization [29].

2.3.3 Other Related Methods

Dynamic programming, a multistage decision process, was introduced by Bellman [30] based on the principle of optimality. It first constructs an optimal policy for the subproblem involving the last stage, propagates the optimal policy to the subproblem involving the last two stages, and continues in this manner until an optimal policy for the entire problem is constructed. Although a typical optimal control technique, it has also been used for trajectory optimization problem over a discrete search space [31] [12].

To find the shortest path, the A^* algorithm [37] or Dijkstra's algorithm ¹

¹Dijkstra's algorithm is also called Label Correcting Methods [35] or Uniform Cost Search.

has been widely used. The idea of both algorithms is to progressively discover shorter paths by exploring nodes in order of increasing value of a cost function. The difference is Dijkstra's uses strictly the accumulated cost from initial state to current state, while the cost function in A^* has one more item, the heuristic estimate of the cost from the current node to the goal.

Besides the methods mentioned above, a group of randomized optimization methods have gained popularity for trajectory optimization or planning. The genetic algorithm (GA) or evolutionary algorithm (EA) [51] models some natural phenomena: genetic inheritance and Darwinian evolution, constitute an interesting category of modern heuristic search. Some applications as [11] and [36] were based on the GA. In [33], when solving the planning coordinated operations among a group of Unmanned Air Vehicles (UAV), the author also investigated evolution-based approaches, which used EA in the underlying optimization problem. Another type of randomized optimization, simulated annealing (SA) [52] [54], a global technique that models metallurgical annealing. It has also been used for applications such as satellite constellation design [40].

Chapter 3

Trajectory Planning in the Longitudinal Plane

As an initial step toward SNI RIA procedure design, we study 2-dimensional longitudinal plane trajectory optimization. To automatically generate a set of candidate low-cost SNI NAPs for final approach to landing, a global optimization method using coupled modified cell decomposition and iterative deepening search algorithms is developed and applied. The search-space is designed to impose realistic constraints on aircraft flight path angle, velocity, and acceleration/deceleration. To find strictly SNI routes, existing fixed-wing traffic corridors are surrounded by a safe separation zone and modeled as impenetrable obstacles.

This chapter begins with the description of the modified approximate cell decomposition and optimal search strategy. After introducing the objective function, the results are presented to illustrate how airspace obstacles, aircraft flight envelopment limitations, and cost function elements influence final approach trajectory shape and corresponding velocity/acceleration profiles.

3.1 Trajectory Optimization Algorithm

SNI final approach trajectory optimization is defined as a single-point (fixed FATO area) boundary value problem in the longitudinal plane, and noise abatement procedures are described by a sequence of one or more segments, each with

constant velocity or acceleration. The approach entry point is constrained by minimum ground distance (e.g. $x = 4000ft$) and altitude (e.g. $y = 2000ft$) so that the optimizer can then define the specific SNI NAP entry fix. The optimization algorithm must minimize cost (noise, time, fuel) in the presence of dynamic constraints and impenetrable airspace obstacles. As discussed previously, techniques for motion planning in obstacle fields, such as roadmap, potential field, and cell decomposition, seem promising. For this work, we adopt a cell decomposition strategy due to its ability to model arbitrary obstacles, guarantee globally-optimal results limited only by discrete cell size, and allow arbitrarily complex cost functions.

The approximate cell decomposition approach was first introduced by [34] and has been utilized in varied forms by a number of researchers. Although typically more computationally complex than local techniques, optimal SNI airspace design benefits more from geometric and cost parameter flexibility than from real-time performance. The fundamental cell decomposition algorithm [19] is given as follows:

Let S (search space) be a rectangloid of R^m , where m is the search space dimension. A rectangloid decomposition of κ of S is a finite collection of rectangloids $\{\kappa_i\}_{i=1,2,\dots}$, such that:

- S is equal to the union of the $\{\kappa_i\}$, i.e.: $S = \bigcup_{i=1}^r \kappa_i$, $i = 1, 2, \dots, r$.

- The interiors of the κ_i 's do not intersect,

$$\text{i.e. } \forall i_1, i_2 \in [1, r], i_1 = i_2 : \text{int}(\kappa_{i_1}) \cap \text{int}(\kappa_{i_2}) = \emptyset$$

Each rectangloid κ_i is called a cell of the decomposition κ of S . Two cells are **adjacent** if and only if their intersection is a set of non-zero measure in R^{m-1} . A

cell κ_i is classified as:

- EMPTY, if and only if its interior does not intersect the obstacle region.
- FULL, if and only if κ_i is entirely contained in the obstacle region.
- MIXED, otherwise.

The **connectivity graph** G associated with a decomposition κ of S is defined as follows:

- Nodes of G are EMPTY and MIXED cells of κ .
- Two nodes of G are connected by a link if and only if corresponding cells are adjacent. In our case the link is a straight line connecting the centers of nodes.

Given a rectangloid decomposition, a **channel** is defined as a sequence $(\kappa_{a_j})_{j=1,\dots,p}$ of EMPTY and/or MIXED cells such that any two consecutive cells κ_{a_j} and $\kappa_{a_{j+1}}$, $j \in [1, p-1]$ are adjacent. An **E-channel** contains only EMPTY cells, while an **M-channel** contains at least one MIXED cell. The most common technique used to build the space is to compute a 2^m -tree decomposition.

3.1.1 Modified Approximate Cell-Decomposition Method

Basic cell decomposition focuses on obstacle avoidance and does not account for additional dynamic parameter constraints (e.g. velocity, acceleration, flight path angle). Modifications to the original algorithm have been made for this work such that constraints can be imposed during the optimization process. In the original

case, with no obstacles, only one cell is generated, and the algorithm will have no results. This implies an obvious solution - a straight line between the initial and final states. However, the solution is not so trivial given dynamic constraints and our multi-parameter objective function, so to find an optimal path without obstacles empty cells are still divided. Given the definition of a decomposition algorithm, the trajectory will be composed of the links between centers of nodes. Cells by default have the aspect ratio of the overall map, and the flight path connects the centers of each cell. A square map yields square cells, which implies flight path angle choices $[45^\circ, 0^\circ, -45^\circ]$ the limits of which violate our $|\gamma| \leq 9^\circ$ constraint. Cell length/width ratio can be used to control flight path angle resolution. For this work, the length:width ratio is set to $100 : 1$, which yields a γ interval of $\sim 0.6^\circ$ as shown in Figure 3.1.

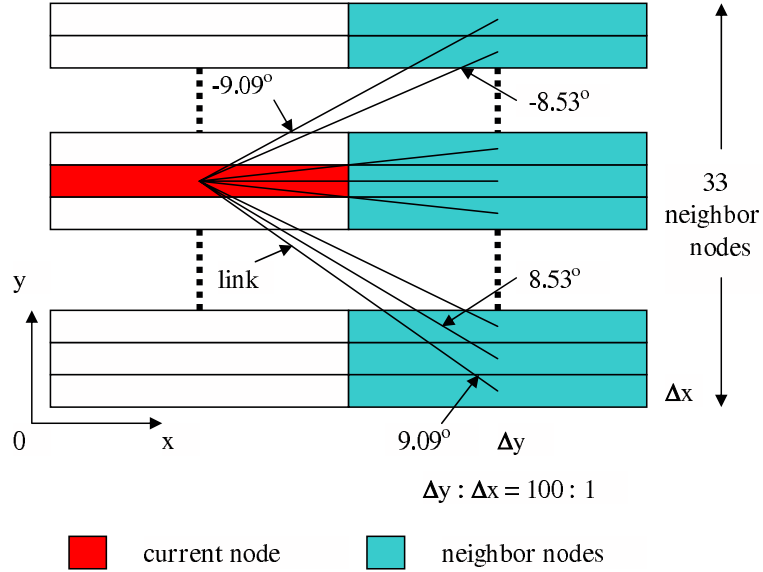


Figure 3.1: Cell Neighborhood Illustrating Flight Path Angle Discretization

Since a rotorcraft is assumed to climb or descend with a flight path angle between $\pm 9^\circ$, the concept of adjacent cells is expanded beyond standard “geomet-

ric” adjacency. Assuming the rotorcraft flies from right to left on the page, 33 nodes to the right of each cell (see Figure 3.1) will be defined as adjacent (neighboring), yielding values $\gamma = \pm 9.09^\circ, \pm 8.53^\circ, \pm 7.97^\circ, \dots, \pm 1.45^\circ, \pm 0.57^\circ, 0^\circ$ for each step. With quad-tree decomposition, all cells maintain the same shape as in the original map. This two-dimensional longitudinal plane, the rootmap, is defined by a rectangle with maximum altitude and lateral distance as its width and length, respectively. The rootmap contains the approach entry region and FATO boundary (landing point) as cell centers, with the landing point defined as the **origin**. To meet the 100 : 1 length:width ratio requirement, the rootmap will be expanded either in length or width. Then, after the quad-tree decomposition step, all cells whose center point is outside the rootmap space are dropped, while the cells with center point in the rootmap are used to construct the search space. Figure 3.2 shows an example modified cell decomposition with a single obstacle.

For existing acceleration constraints imposed primarily for passenger comfort considerations, $(-0.05g \leq \dot{V} \leq 0.05g)$, and the range $38kts \leq V \leq 105kts$ is divided into 20 intervals. Acceleration over a flight segment from node i to k is computed as $(V_k - V_i)/t$, providing a simple approximation to continuous acceleration. To model the discrete velocity value set, each spatial node will become 20 duplicate nodes with different velocities. If the total set of EMPTY nodes within the rootmap space is n , in the modified algorithm $20 \times 33 \times n$ actual search nodes exist.

A final limitation of the original cell decomposition algorithm is that it does not make use of the information in MIXED cells. An optimal path may intersect the MIXED cells without intersecting the real obstacles. The modified cell decom-

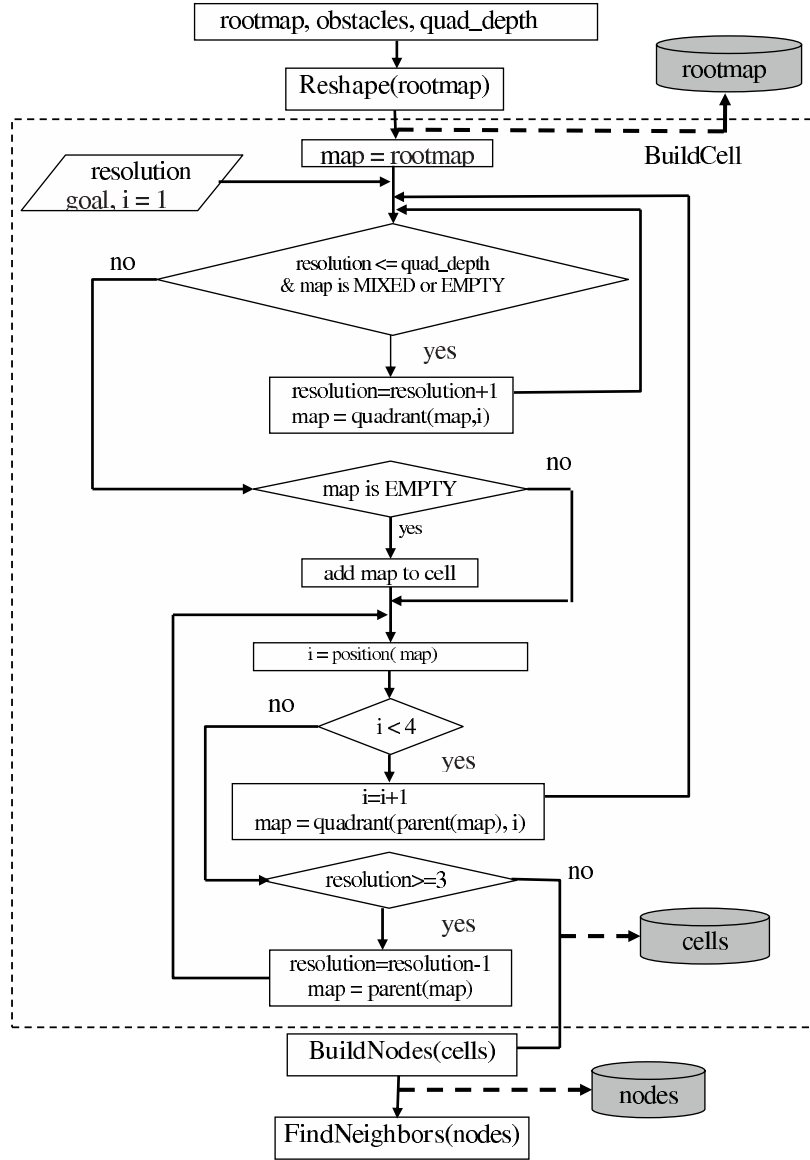


Figure 3.3: **Build** Algorithm

parent cell, the “parent” function returns the parent cell, the “position” function gives the subtree position of the cell with respect to its parent cell, and “node” is a structure containing data for a geometric cell and corresponding velocity value. **BuildCell** is a recursive function to construct the geometric cell set and store the EMPTY cells that reside between the initial and final state “rootmap” vertices to become search nodes. The set of nodes is constructed from cells and is defined by a (cell, velocity) pair plus a list of neighbor nodes. A node j is defined as a “reachable neighbor” of node i if it is geometrically adjacent and meets the γ and acceleration constraints.

3.1.2 Search/Optimization Strategy

Once the cell decomposition map is created, this space must be explored to identify the optimal trajectory given landing site boundary condition (x_f, y_f, V_f) and approach entry constraints (x_i, y_i, V_0) . Given our discrete search space and global optimization requirement, candidate algorithms include dynamic programming and A^* search [37], with an A^* approach selected for this work due to its use of a best-first search to minimize number of expanded search states. In [39], it is shown that dynamic programming typically needs more computational time, since it is less capable of efficiently ordering the search-space. A^* explores nodes in best-first ordering based on an evaluation function $f(n)$. Let $g(n)$ be the actual path cost from the start node (initial state X_i) to current node n , and let $h(n)$ be the estimated cost of the cheapest path from n to the goal. The overall evaluation

function $f(n) = g(n) + h(n)$, and it can be proven that A^* yields an optimal result so long as $h(n)$ is an admissible heuristic (i.e., never overestimates cost from current node to the final state). With $h(n) > 0$, A^* search is “informed” thus is typically more efficient in finding the optimal path. Given the complexity of the cost function, an admissible non-zero heuristic has not been identified thus the trajectory optimizer utilizes $h = 0$ and $g(n)$ set to the cost function described below. When $h(n) = 0$, A^* search emulates Dijkstra’s algorithm or uniform-cost search with evaluation function $f(n) = g(n)$, thus we label our search procedure as Dijkstra’s algorithm. The search process is shown in Figure 3.4. The set of search nodes is constructed during **Build**, where “u” is the current node being expanded, and “c” is the cost from “u” to an adjacent successor node “w”. The current optimal path to each node can be reconstructed from the **parent** list.

The Modified Approximated Cell Decomposition algorithm is a global optimization technique. However, the solution is “optimal” only with respect to the level of discretization when dividing the continuous space into cells. Theoretically, this error can approach zero with infinite quad-tree depth level (resolution); however, computational complexity and path complexity also increases with depth level. We have wrapped an *iterative deepening strategy* [37] around the **Build** and **Search** algorithms to identify optimal solutions for each quad-tree depth level from $1 - n$. Initial low-depth solutions are simple (few segments) but may be costly. Higher-depth solutions approach the globally-optimal cost but will contain numerous flight segments that could only be feasible as SNI NAPs if a trusted autopilot capable of tracking these detailed trajectories is used. The iterative deepening procedure is

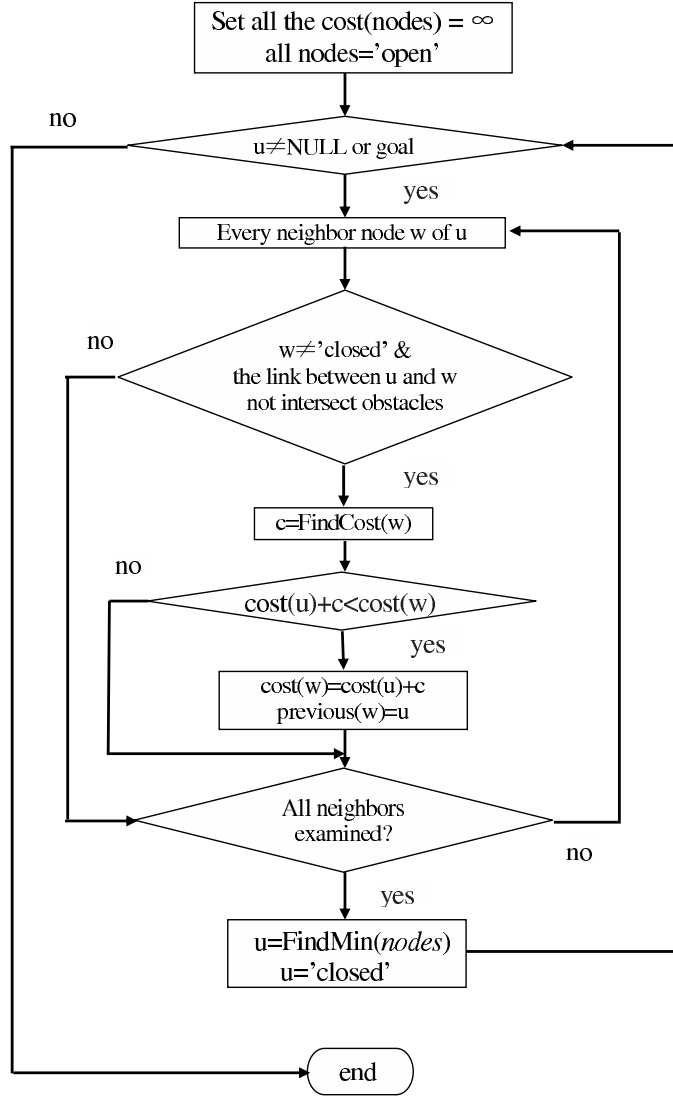


Figure 3.4: **Search** Algorithm

shown in Figure 3.5. In this chapter, we halt at a quad-tree depth where the difference between current and last cost is within a user-defined threshold ϵ . In practice, the pilot/airspace planner can break the loop manually to obtain intermediate solutions obtained from lower quad-tree depth levels.

In this chapter of the dissertation, research is primarily geared toward the *offline* design of NAPs for RIA. For this purpose, the iterative deepening approach provides airspace planners with a solution set, i.e. one solution per quad-tree depth

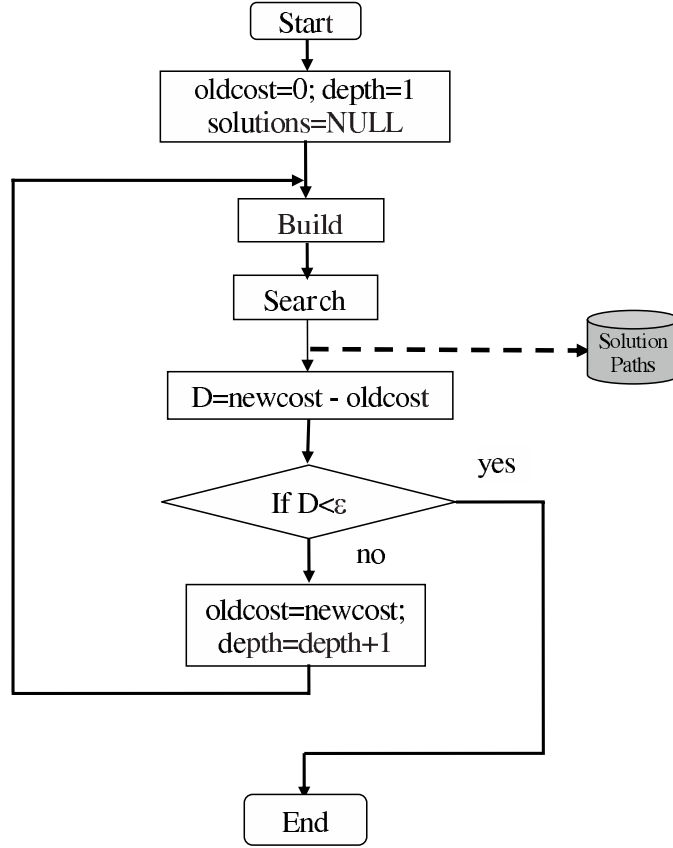


Figure 3.5: Iterative Deepening Strategy

level. The iterative deepening approach, however, can also provide a real-time cockpit SNI NAP planning tool, acting as an anytime [38] algorithm that quickly provides simple (low-depth) routes when necessary but offers lower-cost (high-depth) solutions given sufficient computation time.

3.2 Cost Function

Traditional trajectory synthesis tools permit optimization over fuel and/or time. Pilot or airline preferences and air traffic control constraints contribute to the relative importance (weight) of these two optimization factors. NAP design requires an additional noise cost function term, the relative importance of which can be

varied with time/fuel by varying relative weighting factors. Since fixed-wing airspace “obstacles” are considered impenetrable in this work, they are specifically excluded from the search space rather than modeled in the cost function. As mentioned previously, an AH-1 rotorcraft is used as a representative RIA.

For the SNI airspace design work in this chapter, the cost function $f = g(n)$ for trajectory optimization is given by Eqn. 3.1:

$$f = c_1 \int_{t_0}^{t_i} t_0 10^{P/10} dt + c_2(t_i - t_0) + c_3 m_{fuel,i} + c_4 \dot{\gamma} \quad (3.1)$$

In this expression, t_0 is the initial state time, and t_i is the current time at state i in Eqn. 2.1. BVI noise, given by the empirical expression for P in Eqn. 2.4, is integrated over the flight path in Eqn. 2.5. Transient maneuvers resulting in non-zero $\dot{\gamma}$ and acceleration \dot{V} are governed by vehicle dynamics and will affect all cost function terms in Eqn. 3.1. ($\dot{\gamma}$ over a transient maneuver from segment i to k is computed as $|\gamma_i - \gamma_k|/\Delta t$. In our case, Δt is set to 1 second.) Because $\dot{\gamma}$ also reflects passenger comfort and is not specifically considered in our quasi-static BVI noise model, we include a distinct $\dot{\gamma}$ cost term. Coefficients c_1 , c_2 , c_3 , c_4 may be adjusted based on relative prioritization of time, fuel, noise, and rejection of flight path excursions $\dot{\gamma}$.

3.3 Sensitivity Analysis

For all presented examples, the FATO area boundary is defined as $x = 50ft$, altitude $z = 10ft$, and the approach entry area is constrained by $x > 40,000ft$ and

$z > 2,000ft$. To remain within the valid Q-SAM region from which our noise model was derived, a minimum altitude constraint $z > 50ft$ was also imposed. Presented results are organized to disambiguate the effects of each cost term as well as search-space discretization (quad-tree depth) level. Unless otherwise stated, presented results utilize a quad-tree depth level of 6, generally observed to provide sufficient optimization resolution without introducing unreasonable path complexity.

3.3.1 Noise Only

Typical of a helicopter approach, consider a case in which initial velocity is 95 knots and final velocity is 45 knots. The resulting optimal “bang-bang” solution is composed of climbs and decelerating descents as shown in Figure 3.6. This result maximizes the distance of the wake from the rotor, thereby minimizing vortex-induced noise. Figure 3.6 also illustrates the worst-case (maximum noise) approach, during which wake and rotor blade are minimally separated. The difference between optimal and worst noise cost is more than 20 dB, and the path shape for worst-case noise trajectory is similar to a standard shallow-descent landing procedure.

In practice, certain residential communities may be more noise sensitive than other commercial use or unpopulated regions. In this case, cost function weight c_1 can be assigned such that noise-sensitive segments incur higher penalty than less-sensitive areas. Consider an example in which a noise-sensitive area exists under the final approach path between $\sim 25,000$ and $32,000ft$ from touchdown. With higher weight on this segment, the optimal solution (Figure 3.7) includes an accelerating

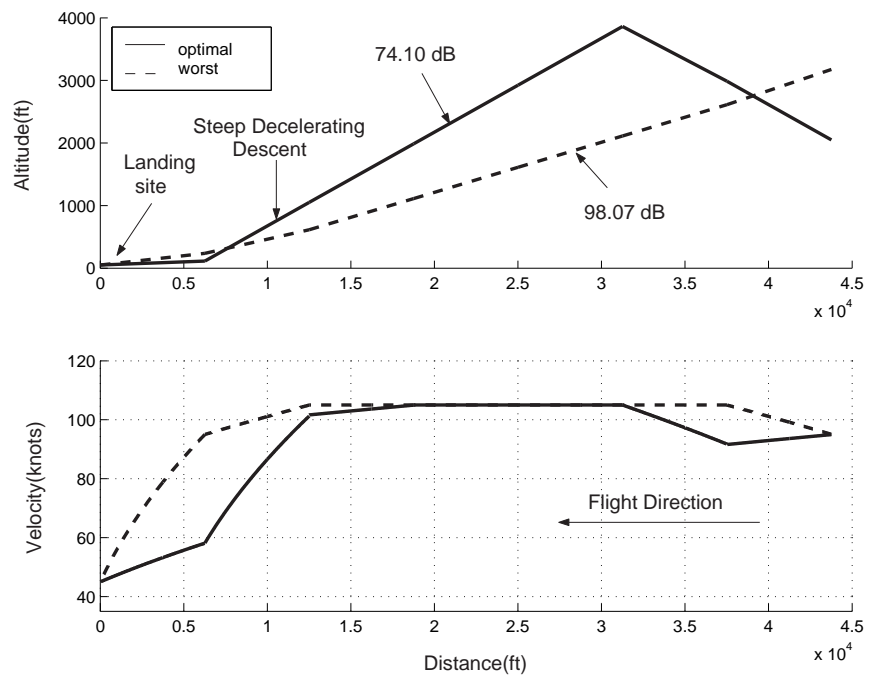


Figure 3.6: Noise Optimal and Worst Approaches from 95 knots to 45 knots

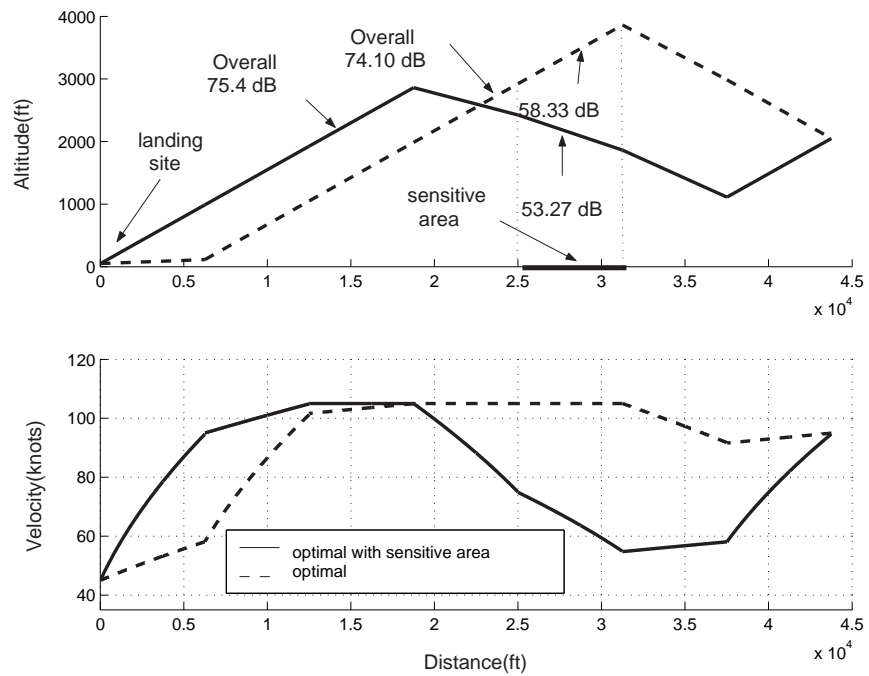


Figure 3.7: Noise Optimal Approach with Extra Noise Penalty over Sensitive Region

climb during the noise-sensitive segment, generating $\sim 5dB$ less noise per unit time than the noise-optimal solution with uniform c_1 .

3.3.2 SNI Noise-optimal Approaches

The SNI NAP optimization procedure was next applied with a single intersecting airspace obstacle to illustrate the effects of intersecting fixed-wing airspace corridors. This obstacle is modeled as a polygon that approximates the perpendicular intersection of the longitudinal SNI final approach plane with a cylindrical three-dimensional fixed-wing airspace corridor of radius 300 ft. As illustrated by Figure 3.8, the globally-optimal solution lies within such an obstacle, and the resulting solution is the minimum of alternative local minima from the unobstructed longitudinal plane or neighboring sub-optimal solutions adjacent but exterior to the obstacle. The particular choice depends on the nature of the objective function in the neighborhood of the optimal solution. If the objective function remains relatively constant when perturbed about the optimal solution, a neighboring sub-optimal solution may be preferred. Otherwise, one of the numerous other locally-optimal solutions that exhibit large flight path excursions and do not intersect the obstacle would be selected. Additionally, from Figure 3.8, it is noted that the cost difference is only 0.01 dB, which is a very tiny difference. That shows the introduction of a moderate obstacle set does not affect the final optimal cost significantly.

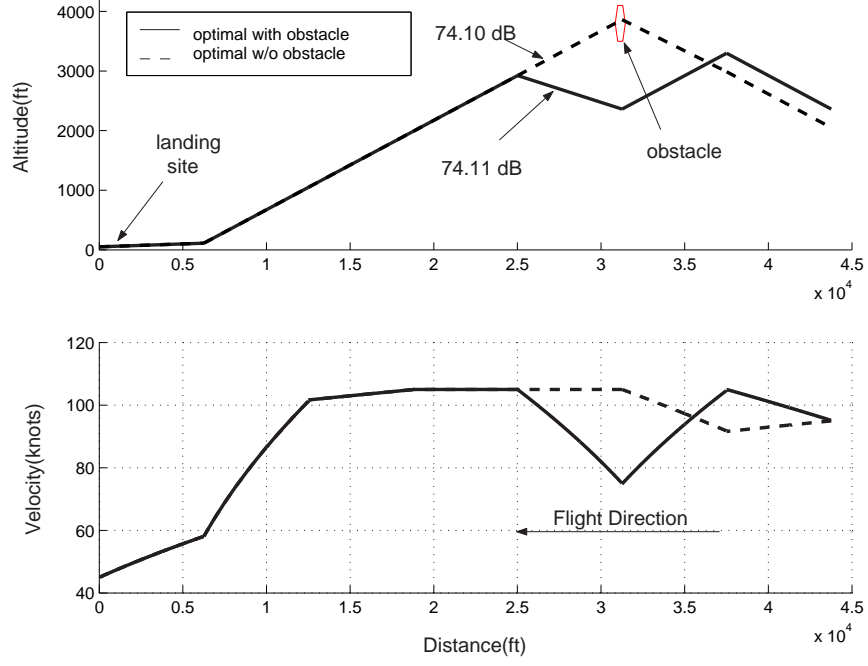


Figure 3.8: Optimal SNI Approach with A Single Airspace Obstacle

3.3.3 Effect of Longitudinal Path Angle Changes

Figure 3.9 shows the noise-optimal solution with additional penalty on $\dot{\gamma}$ imposed for passenger comfort and because the trajectory model used in this chapter does not include additional noise generated by $\dot{\gamma}$ transitions. In this case, the quad-tree depth level is increased to 7, and c_4 is set to a high value to approximately balance the numerically-large noise level (in pressure) with $\dot{\gamma}$. (Due to the discrete search space, the optimal solution is influenced only when significantly increasing c_4 .) The c_4 values listed in this case were chosen to impact but not dominate overall cost. After exploring a series of relative cost weights, coefficient c_4 is set to 2.5×10^5 . Compared with the baseline - global optima, the resulting solution reduces flight path excursions with one less “bang-bang” transition between extrema, while the noise has increased only 0.03 dB. If c_4 is further increased to 1.7×10^6 ,

the path has only a single bang-bang transition with an additional 0.05 dB. Thus, it has been shown that moderate longitudinal path angle changes do not increase the final cost significantly.

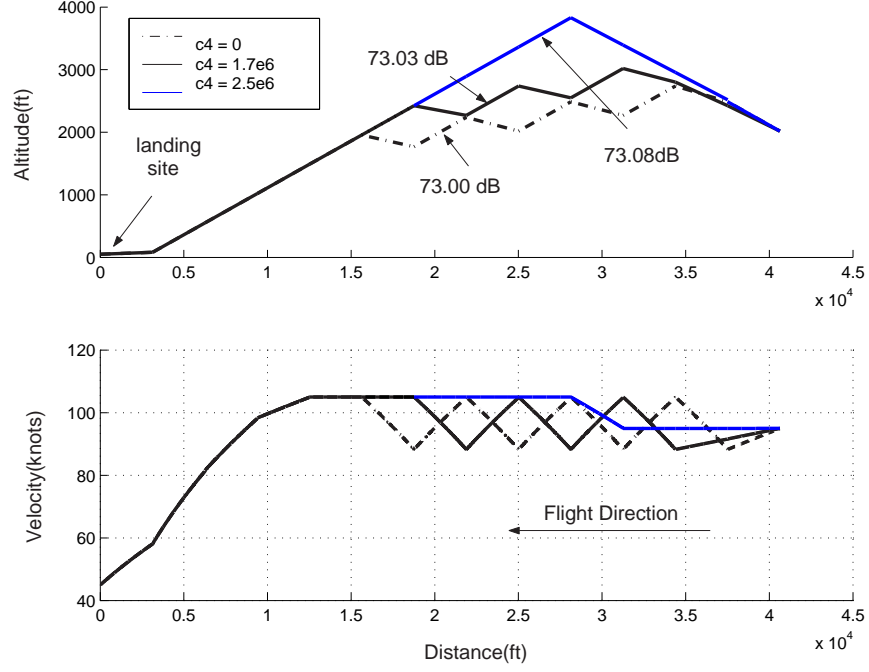


Figure 3.9: Approach From 95 kts to 45 kts For Varied c_4

3.3.4 Time/Fuel Optimal Solutions

As expected, approach duration primarily depends on aircraft speed, with secondary effects from flight path angle transitions that perturb trajectory length. Quite simply, a time cost penalty drives the optimizer to select the highest possible approach speed, while a fuel penalty will drive an optimal path to follow a “best-glide” speed during approach and also favors the maximum descent angle for gravity-assist speed maintenance. However, since the case study from this work enforces maximum and minimum speed constraints of 105 knots and 38 knots, respectively,

both time and fuel penalties drive the solution toward a maximum-velocity, best-glide trajectory from initial to final states, as shown in Figure 3.10. Under the velocity constraints the optimal result first accelerates to the maximum possible speed (V_{max}), which is maintained until a final maximum-deceleration ($-0.05g$) maneuver matches final state velocity (45 knots). As a comparison, Figure 3.10 also depicts the worst case, in which an initial climb and deceleration to minimum velocity result in maximum fuel and time cost. If an airspace obstacle constraint is imposed along the optimal path, the resulting solution, a global minimum for the specified boundary constraints, is an adjacent solution with maximum speed and overall descent angle.

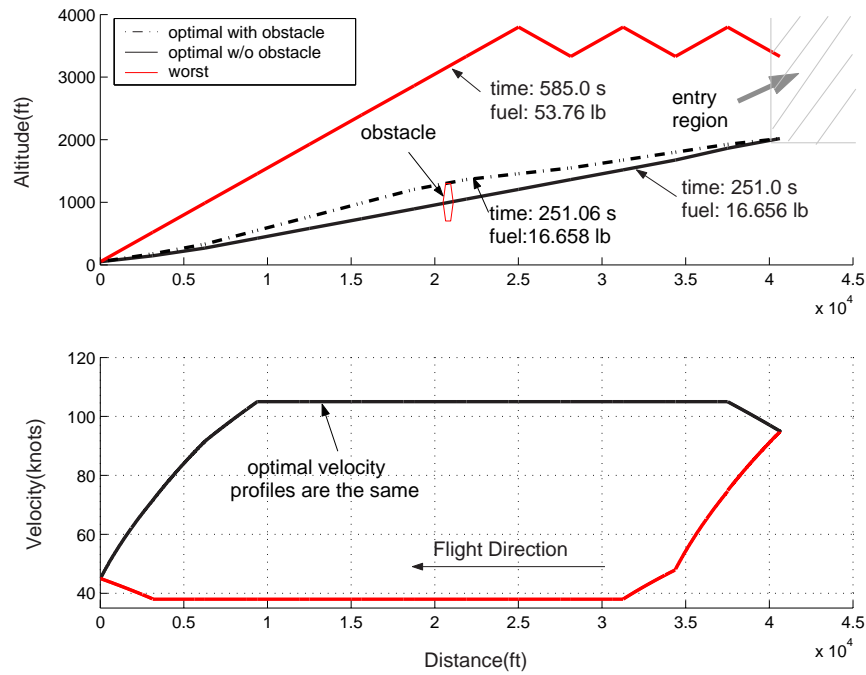


Figure 3.10: Minimum and Maximum Cost Based on Time/Fuel Cost

Table 3.1: Optimal Solution Cost as a Function of η

Case	η	Noise(SEL_{av})(dB)	Flight Time(s)	Fuel(lb)
A	0	73.0	275.42	18.25
B	10^6	73.57	259.81	16.28
C	10^7	76.67	252.59	16.47
D	10^{10}	90.03	251.10	16.66

3.3.5 Noise and Time/Fuel Trade Study

From previous discussions, it may be observed that the optimal time/fuel path is similar to the worst-case shallow descent trajectory for BVI noise, while the optimal BVI noise trajectory increases time and fuel usage. The final SNI NAP solution must therefore balance noise with time and fuel. To simplify this tradeoff and gain insight for the final approach case studied in this chapter, the cost function is further simplified as Eqn. 3.2, with $\eta = c_2/c_1$. Because the time and fuel optimal solutions are similar given the approach velocity constraints, only the simple flight time cost is incorporated. To disambiguate the tradeoff between noise and time/fuel cost, the effect of $\dot{\gamma}$ is not included.

$$f = \int_{t_0}^{t_i} t_0 10^{P/10} dt + \eta \times (t_i - t_0) \quad (3.2)$$

Figure 3.11 illustrates optimal path cost as a function of η , while Table 3.1 provides the corresponding cost values. It may be observed that, since the minimum-noise path has high velocity even though the path profile appears worst-case for time/fuel, the time/fuel actually consumed is significantly better than the real worst-case. However, there is still a significant efficiency difference, particularly

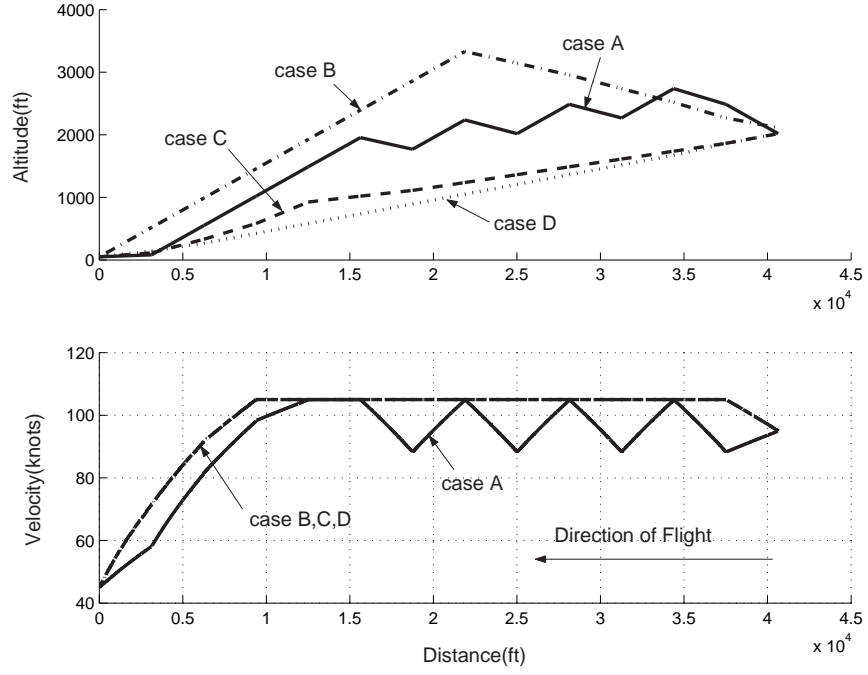


Figure 3.11: Optimal Approaches with Varied η

if integrated over numerous flight operations. Figure 3.12 illustrates the tradeoff between a low-noise, high time/fuel solution (low η) and a high-noise, low time/fuel solution (high η). As shown in the figure, noise cost is primarily minimized until η increases to $\sim 10^6$. Within the interval $[10^7, 10^{10}]$, dominance transitions until the time cost term drives the optimization process. By selecting a coefficient η within this transition region (e.g., highlighted cases B or C in Table 3.1), a solution acceptable in noise, time, and fuel may be identified. Note that the time and fuel cost terms are similar but not identical, as illustrated by minimum fuel use in Table 3.1 for $\eta = 10^6$ rather than higher values. This supports the presence of both time and fuel terms in the full cost function in Eqn. 3.1.

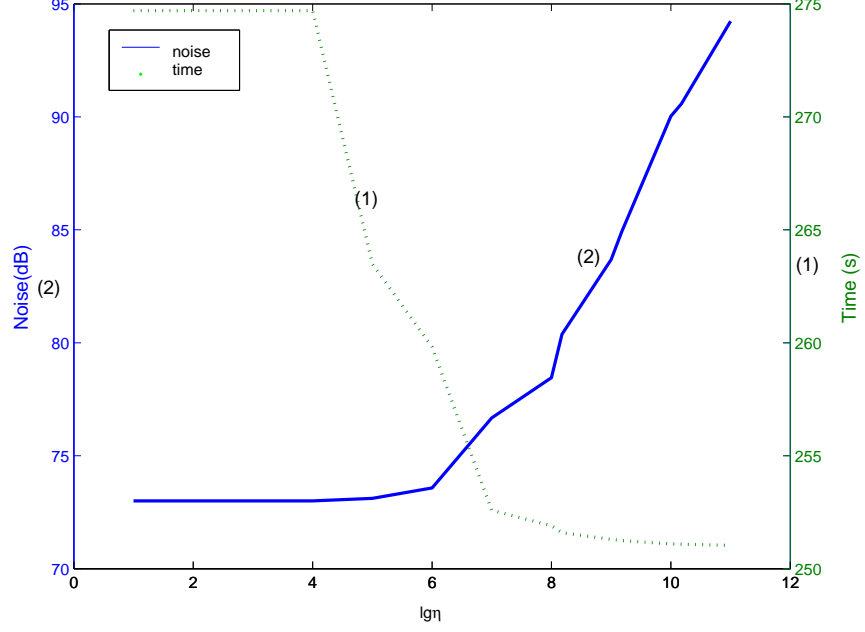


Figure 3.12: Optimal Trajectory Noise and Time as a Function of η

3.3.6 Iterative Deepening Solution Set

Use of the iterative deepening search strategy provides a set of optimal solutions, one for each quad-tree depth level. Figure 3.13 provides an example solution path set with only the noise cost term, while Table 3.2 provides quad-tree depth, total number of flight segments, average completion time when executed under Linux on a 1 GHZ Pentium III, and NAP cost for each of the five generated solutions. Figure 3.13 illustrates the increasing trajectory complexity as depth is increased. For Case 1 (depth = 3), a single-segment solution is identified. Note that the high cost results from the constant deceleration from 95 to 45 knots that requires the trajectory to pass through the BVI peak noise region. Case 2 extends well beyond the 40,000 ft distance constraint, leveraging an initial accelerating climb to reduce overall noise. Cases 3 and 4 couple a single accelerating climb with a decelerating

Table 3.2: Noise-optimal Solution Cost with Increasing Depth Level

Case	Quad-Tree Depth	Flight Segments	Computational Time(s)	Ground Noise (dB)
1	3	1	$\ll 1$	88.60
2	4	3	3	80.27
3	5	4	8	76.59
4	6	7	146	74.10
5	7	13	1756	73.00

descent, providing reasonable 74-76 dB NAPs with moderate complexity. Case 5 minimizes noise with thirteen flight segments, providing a low-cost but undesirable “bang-bang” flight trajectory. The Case 1-3 solution set is typically generated in less than 20 seconds, providing a potential real-time noise abatement flight trajectory-planning tool that could be implemented in the cockpit even before the RIA concept is realized.

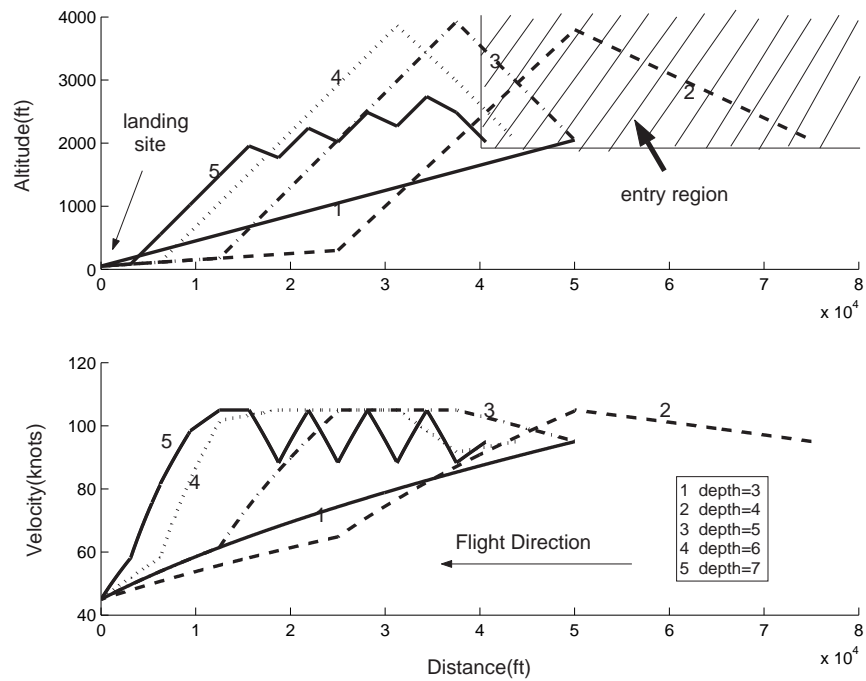


Figure 3.13: Noise-optimal Approaches as a Function of Quad-tree Depth

3.4 Summary

In this chapter, an approach for optimizing 2-dimensional segmented SNI NAP for RIA has been presented. Fixed-wing airspace corridors are treated as impenetrable obstacles, and trajectories are optimized with respect to radiated noise, fuel, and time. The approximated cell-decomposition method has been modified to introduce dynamic constraints and small intervals for design variables. The new method is combined with Dijkstra’s search algorithm to generate an optimal solution.

Optimal final approach trajectories for an AH-1 helicopter are generated using an empirical model of BVI noise derived from the test-validated Q-SAM. These trajectories exhibit several interesting characteristics. A noise-optimal rotorcraft trajectory uses accelerating climbs and decelerating descents to the landing site at maximum $|\gamma| = 9^\circ$ to avoid the peak BVI noise region. Airspace obstacles force the optimal SNI trajectory to a sub-optimal solution exterior but adjacent to the obstacle or, in some cases, to a very different path corresponding to a local minimum from the unobstructed longitudinal plane. By comparing the optimal noise costs calculated with and without obstacles, we find this optimization problem has many local optima. A $\dot{\gamma}$ penalty smooths the trajectory, which otherwise is a bang-bang solution to minimize noise alone. Increased noise cost penalty over noise-sensitive communities biases the trajectory to perform an accelerating climb over the sensitive region to reduce BVI noise. Additionally, a clear tradeoff between noise and time/fuel efficiency exists, enabling the user to balance the relative weighting factors to build a low-noise, efficient trajectory.

The iterative deepening search strategy serves two purposes: it provides an anytime algorithm for approximate solution computation in real-time, and it generates progressively more complex segmented trajectories that can be compared with the simpler solutions to tradeoff trajectory complexity with reduction in noise, time, and fuel cost.

Chapter 4

Terminal Area Trajectory Planning in Three Dimensions

In this chapter, we extend the trajectory optimization model to 3-D, incorporating population data into the objective function, and focus on a practical case study: the design of SNI RIA approach procedures to Baltimore-Washington International (BWI), a rapidly growing urban airport. Figure 4.1 shows the models and algorithms we integrate to build optimal segmented SNI trajectories. It is shown that, to solve the problem, two groups of information must be modeled: airport-related data and RIA-related data.

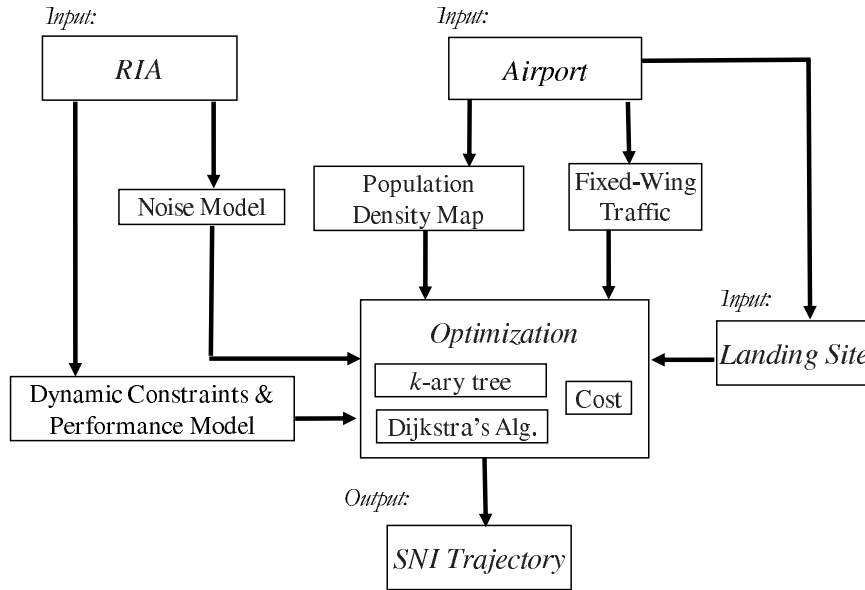


Figure 4.1: Optimization of Segmented SNI Trajectories.

This chapter begins with a description of our discrete optimization method and the objective function used to generate SNI NAPs. For Baltimore-Washington

International (BWI) airport, a RIA landing site is identified, and population data is compiled from 2000 census data. To find strictly SNI routes typical BWI East-flow traffic operations are surrounded by safe separation zones and modeled as impenetrable obstacles. Candidate AH-1 NAPs are presented to illustrate how population density, aircraft flight envelope limitations, and entry region influence final trajectory shape and corresponding velocity profiles.

4.1 Trajectory Optimization Algorithm

In the previous chapter and in [44], we utilized a modified approximate cell decomposition strategy that could model arbitrary obstacles, guarantee globally-optimal results limited only by discrete cell size, and allow arbitrarily complex cost functions f . In 3-D, however, it is difficult to manage computational complexity when sufficient resolution is present in the discretized cell map. To better manage complexity, we now represent the search-space as a k -ary tree, defined as a search space with no more than k children for each node. As a preliminary to k -ary tree construction, we impose dynamic parameter constraints well within limits of both the vehicle performance envelope and passenger comfort. For longitudinal flight, limits are placed on flight path angle γ and acceleration \dot{V} , with the addition of parameter β for 3D to represent the change in heading between flight segments. The goal with our algorithm and this simple parameter space is to facilitate global optimization over experimentally-verified noise and performance models. In this chapter, we assume the transition between flight segments is instantaneous and

incurs no additional cost, then study the impact of this simplification in the following chapters.

Given these dynamic parameter restrictions, we use four control/design variables γ , \dot{V} , β and Δd , where Δd is the distance step in the lateral (xy) plane and along the direction of radius of the circle whose origin is located at the touchdown site. Δd is defined by number of segments through:

$$\Delta d = \frac{\text{ground projection of distance between approach and land sites}}{\text{number of trajectory segments}} \quad (4.1)$$

Note that the set of control variables is reduced to three variables γ , \dot{V} and Δx when restricting solutions to 2D (longitudinal plane) only. For 2D and 3D cases, control variables are discretized and the combinations over these variables form an upper bound of the branching factor for our k -ary tree.

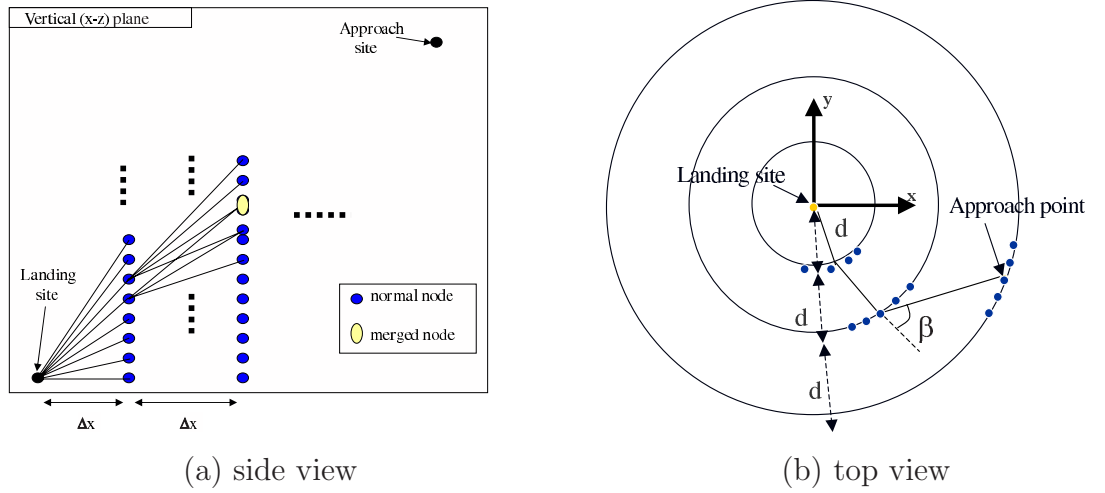


Figure 4.2: k -ary Tree Decomposition and Merging (spatial view)

We adopt an incremental search method in which the search tree is expanded until the final solution is found. At each iteration, the next available k -ary tree node

is selected by the search algorithm in order of increasing cost, as discussed below. When exploring a k -ary tree node, k candidate “child” nodes are generated. To minimize search-space size, new nodes are merged with existing nodes if they are in a small “error box” which denotes approximately the same velocity and spatial position in which geometric (x,y,z) coordinates match to within a specified tolerance. This decomposition and merging strategy is shown in Figure 4.2 for the spatial parameters without control variable V . For two-dimensional optimization (e.g., an approach in the longitudinal plane), our previous cell decomposition and new k -ary tree strategies are comparable in representational ability and efficiency. For 3-D trajectories, the k -ary tree enables more accurate representation, and node merging efficiency improves significantly. The pseudocode for the k -ary tree decomposition strategy is shown below in procedure *TreeBuild* (see Figure 4.3), where M, N and R are the resolutions of γ , β , and V , respectively. In this chapter, these intervals are set as $\Delta\gamma = 1^\circ$, $\Delta\beta = 4^\circ$ and $\Delta V = 3 \text{ knots}$. During expansion of a node *enode*, the procedure identifies all child nodes and inserts them into the node stack if they are not merged and if the connecting path from *enode* does not intercept any fixed-wing airspace obstacles. These nodes are marked as the neighbors of *enode* and their tree level (segment number) is incremented relative to *enode*.

In this chapter, we study and analyze the feasibility of this modeling strategy and its application to design SNI NAPs at a major urban airport (BWI). We again adopt Dijkstra’s algorithm as the search strategy and uncover issues with its computational complexity.

```

Procedure TREEBUILD(enode, nodes, obstacles)

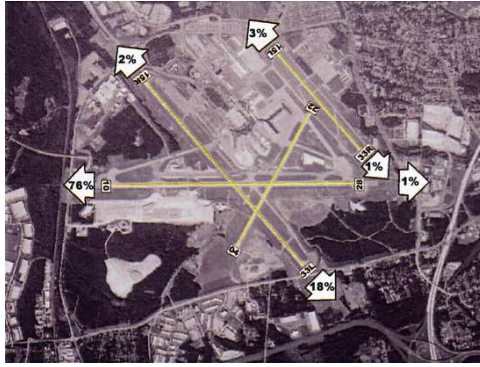
global  $\gamma[M], \beta[N], V[R], \Delta d$ ;
Node nnode;
begin
  for each combination of  $\gamma[i], \beta[j], V[k], \Delta d$  do
    (x, y, z, V[k])  $\leftarrow$  CalculateState( $\gamma[i], \beta[j], V[k], \Delta d, enode$ );
    nnode  $\leftarrow$  new Node (x, y, z, V[k]);
    nnode.level  $\leftarrow$  enode.level + 1;
    if not Merge(nnode, nodes) and
      not Intercept(nnode, obstacles) do
      push nnode into stack enode.neighbor;
      nnode.cost  $\leftarrow \infty$ ; nnode.status  $\leftarrow$  'open';
      push nnode into stack nodes;
    end
  end
end

```

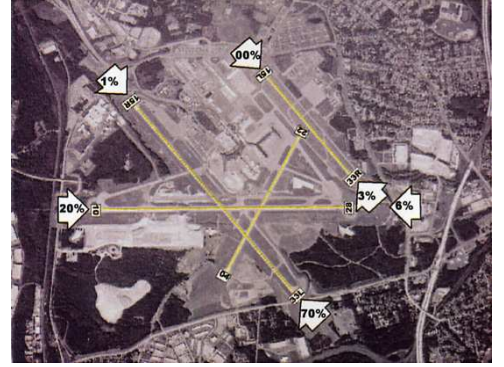
Figure 4.3: *TreeBuild* Algorithm

4.2 Terminal Area Environment Models

To assess the applicability of our SNI NAP design procedures, we studied the design of SNI approaches to Baltimore-Washington International airport. As a major airport hub for Southwest Airlines, flight operations at BWI have steadily increased over the past years. BWI also is situated in populous area with minimum expansion space. We studied traffic statistics to define a practical RIA landing site. As shown in Figure 4.4, short runway 15L/33R is actually used rarely, representing 3-4% of all 2003 operations [45]. Thus, as a candidate for all classes of RIA, we define 15L/33R as our RIA landing site.



(a) Departure



(b) Approach

Figure 4.4: Annual Operations at BWI Airport

4.2.1 Fixed-Wing Traffic Obstacle Model

To explore strictly SNI noise-optimal trajectories, flight track data was acquired for BWI airport. Daily approach and departure tracks for the BWI terminal area can typically be grouped into either east-flow or west-flow, depending on prevailing wind direction. In our work, track data for all arrivals and departures on

July 3, 2003 was utilized as a sample for east-flow operations.

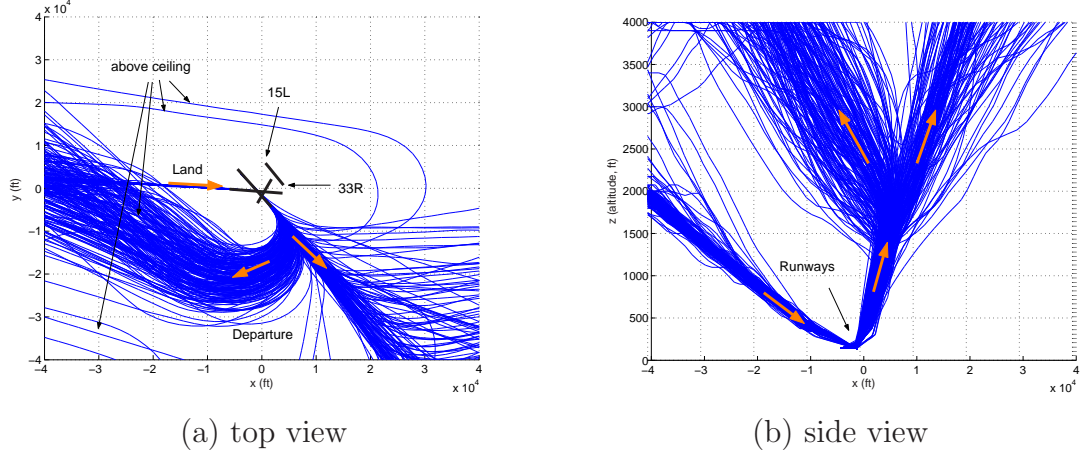


Figure 4.5: East Flow Operation at BWI Airport, July 03, 2003

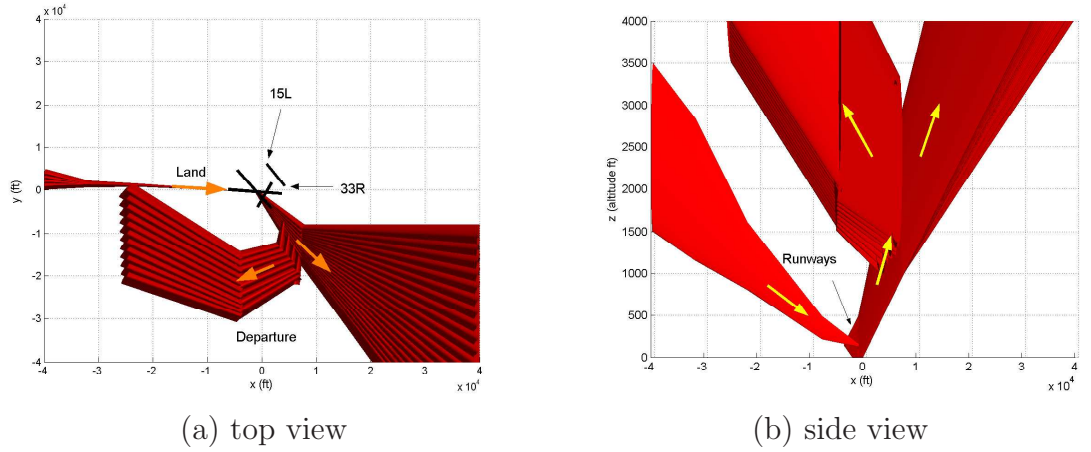


Figure 4.6: Obstacles Modeled for East Flow Operation

Figure 4.5 shows track data top and side views and also presents the BWI runway layout. As previously discussed, runways 33R/15L are statistically much less used than other BWI runways, representing 3-4% of all 2003 operations [45] Given also that when used, 33R/15L currently support smaller general aviation/business jet traffic, so existing operations utilizing this short runway can be removed from consideration without significant impact on passenger throughput.

To represent fixed-wing routes as impenetrable obstacles, we wrapped the existing east-flow fixed-wing flight tracks in a set of cylinders and cones to define no-fly areas for SNI operations. For our case study, traffic obstacles are modeled below $4,000ft$, a ceiling also placed on RIA approach trajectories. As a comparison with flight track data, top and side views of our obstacle field are shown in Figure 4.6. The set of obstacles are enlarged to enforce a $1000ft$ clearance constraint and treated as impenetrable. To support the east-flow traffic model, the new RIA traffic is constrained to land near the approach end of runway $15L$, parallel to fixed-wing traffic departing from $15R$.

4.2.2 GIS Factor - Population Model

Since segmented trajectories must be designed to minimize noise over populous areas, we employ a cost function in which noise propagated to the ground is weighted by population density. BWI area population data was downloaded from the year 2000 census [46]. By using standard geography software *ArcView GIS 3.2*, we obtained a population density distribution for the area around BWI.

Figure 4.7 shows the population distribution around BWI. The blocks with darker red denote high population density areas, while lighter red represents low density. For this work, only the blocks within 7 miles of BWI airport are taken into account, as illustrated in Figure 4.7. We define noise “cost” as the sum of the noise experienced over a set of ground-based microphones, where the noise at each microphone is weighted by the population density of the region surrounding

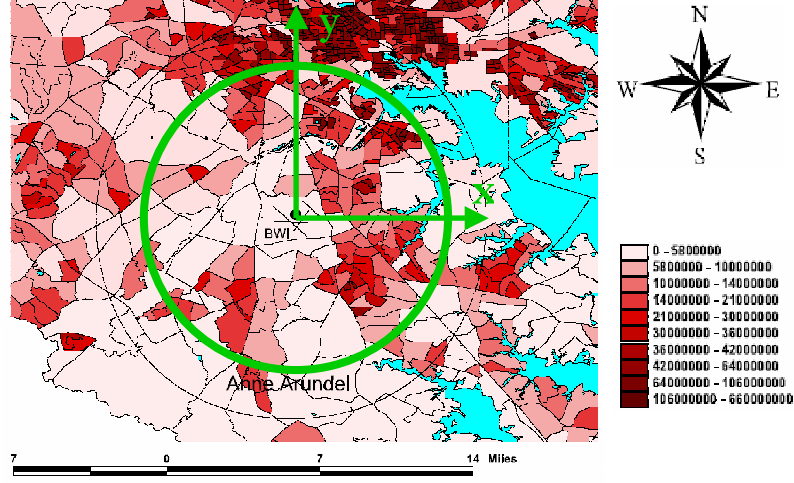


Figure 4.7: Population Distribution Map Around BWI Airport

that microphone. Evenly spaced microphone locations (Figure 4.8(a)) capture most information, but such a design may also neglect the small dense-population blocks shown in Figure 4.7. To prevent this, a large number of microphones must be modeled and computational cost will be prohibitive. In this work, to get an accurate representation of noise in reasonable computational time, we define the center points of the Figure 4.7 population blocks as “microphone locations” for a ground noise measurement matrix over which noise is normalized (see Figure 4.8(b)). Since, intuitively, the noise annoyance should be worse when the air vehicle flies over the densely-populated areas, we set the importance (weight) of each block based population density instead of the total population. Thus, cost function penalty weight for each “microphone” is proportional to the population density at that site. Figure 4.7 also shows the position and orientation of the inertial reference coordinate system, in which the z coordinate coincides with altitude.

Combining Q-SAM noise data with population density, total noise energy over

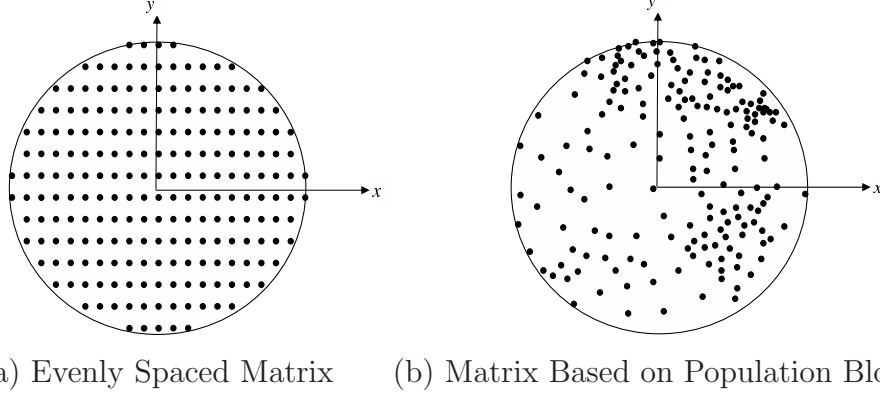


Figure 4.8: Population Distribution Map (BWI) and Measurement Matrix

all K 'microphones' for a single trajectory segment is computed as:

$$N_{i,i-1} = \frac{\sum_{k=1}^K 10^{\frac{P_{i,i-1}}{10}} \times W_k}{\sum_{k=1}^K W_k} \quad (4.2)$$

where each W_k is the population-based weighting factor for 'microphone' k and $P_{i,i-1}$ is retrieved from the Q-SAM database as stated in Eqn. 2.3.

4.2.3 Objective Function

In this chapter, we focus on the optimization over population-weighted BVI noise, which is the key factor for low-altitude approaches. Total cost over all n trajectory segments is:

$$J_n = \sum_{i=1}^n N_{i,i-1} \times t_{i,i-1} \quad (4.3)$$

where $t_{i,i-1}$ is the duration for the single trajectory segment between boundary nodes i and $i - 1$, and $N_{i,i-1}$ is the normalized noise from Eqn. 4.2. The optimization goal given our iterative deepening strategy is then to progressively find solutions

for $n = 1, 2, \dots, n_{max}$ defined as trim flight segment sequences that minimize cost J_n . Besides all constraints in the previous chapter, we artificially incorporate the lateral angle constraints $-80^\circ \leq \beta \leq 80^\circ$ to limit the magnitude of heading changes between adjacent trim flight segments.

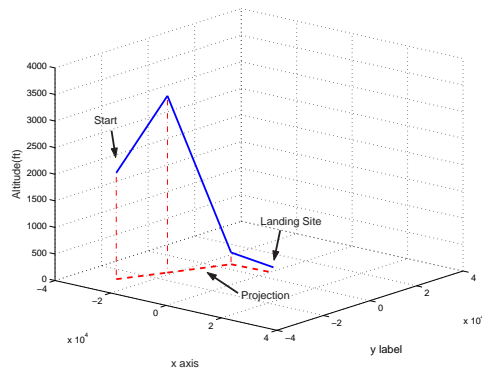
4.3 Simulation Results

Based on the AH-1 and BWI models presented above, we now study the design of noise-optimal SNI approach trajectories for BWI airport. For all approach cases, the initial velocity is prescribed as 95 knots and final velocity 45 knots, typical values for an AH-1. The trajectory spans the area within 7 miles (approximately 40,000 *ft*) from BWI and to a ceiling of 4,000 *ft*, while the lowest altitude is 100 *ft*, given that the Q-SAM noise model is only valid above 100 *ft* and that population is not dense near the touchdown site. For distinction, “dB-PD” is defined with units of $10\log J_n$, which represents the population-density weighted noise cost over the area within a 7 mile radius of BWI.

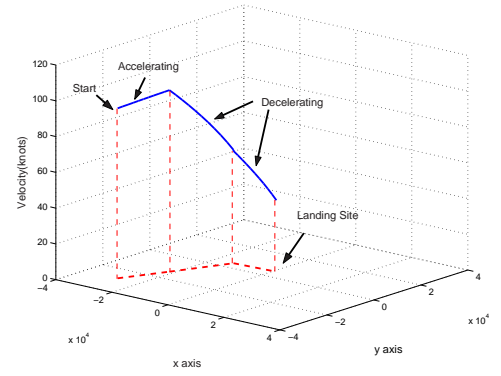
4.3.1 Three-Dimensional Optimal Trajectories

For this case, the 3D approach problem is defined from start state ($x = -28,000$ ft, $y = -28,000$ ft, $z = 2000$ ft, $V = 95$ knots) to final state ($x = 1000$ ft, $y = 5000$ ft, $z = 100$ ft, $V = 45$ knots) approaching the Runway 15L threshold, which means the rotorcraft flies from the southwest approach corner to the landing site.

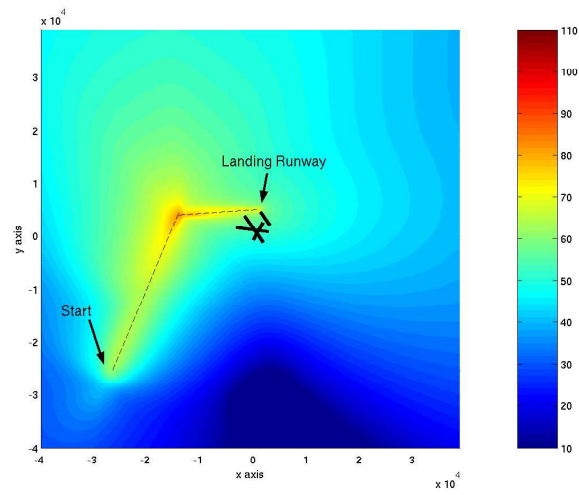
Optimal solution behavior is similar to 2D cases. Figure 4.9 shows the three-



(a) Path Shape



(b) Velocity Profile



(c) BVI SEL Distribution (Top View)

Figure 4.9: 3-Dimension 3 Segment Noise-Optimal Trajectory

Table 4.1: Comparison With Varying Segment Number in 3D

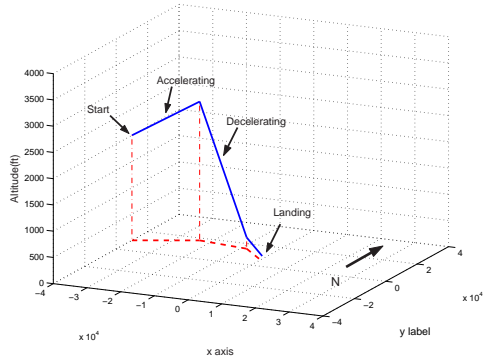
Seg. No. Comp. Time Cost	2 42 sec 57.7 dB-PD	3 44975 sec 48.2dB-PD
Path		
Velocity		
SEL Map		

segment solution path and velocity profiles and the ground noise distribution (SEL). Table 4.1 illustrates the tradeoff between path complexity (represented as number of flight segments) and noise cost. Note, however, the rapid increase in optimization execution time as number of segments increases due to the increase in search-space size as the k -ary tree is built to a greater 3-D depth. Although there is a computational bound on the total number of 3-D flight segments that can be explored, we do not view this as a significant limitation since neither pilot nor air traffic controller will want to manage an approach procedure with more than 3-5 distinct low-altitude segments. In this chapter, our goal is to build SNI procedures that can be studied and adopted as standard for each RIA landing site, favoring identification of globally-optimal results over real-time computation.

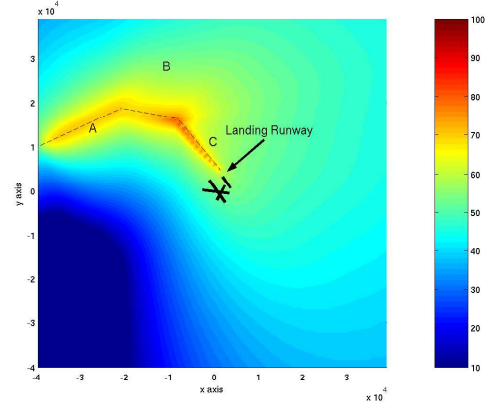
4.3.2 Effects of GIS Factor - Population

Previous results were optimized over the cost function from Eq. 4.3 that included population-based weighting factors. To investigate the effects of population density weighting, we set all population weights W_k to 1.0 (uniform density distribution) and re-computed a representative three-segment 3-D approach. Consider an approach with start state ($x = -40,000$ ft, $y = 10,000$ ft, $z = 2000$ ft, $V = 95$ knots) to final state ($x = 1000$ ft, $y = 5000$ ft, $z = 100$ ft, $V = 45$ knots) landing on Runway 15L.

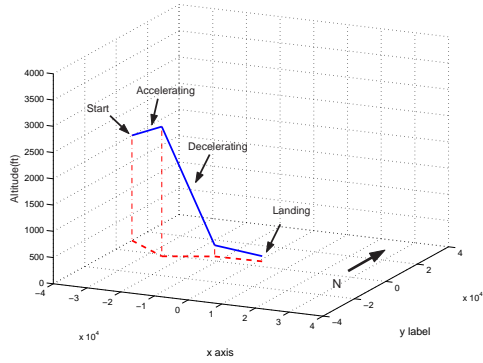
A comparison of population-weighted and uniform cost weight trajectories is provided in Figure 4.10. Figure 4.10(a) and 4.10(b) show the case with no effect of



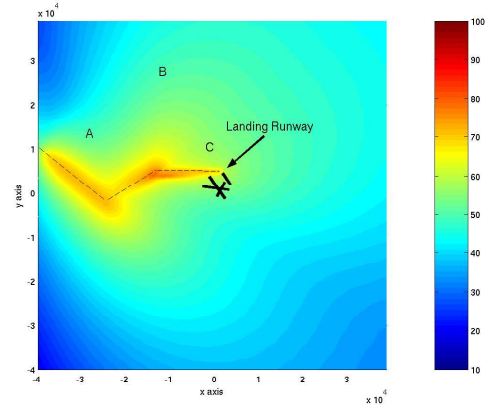
(a) Optima W/O Population



(b) SEL Distribution ($W_k = 1.0$)



(c) Optima With Population



(d) SEL Distribution

Figure 4.10: Effects of Population Density on Noise-Optimal Solution

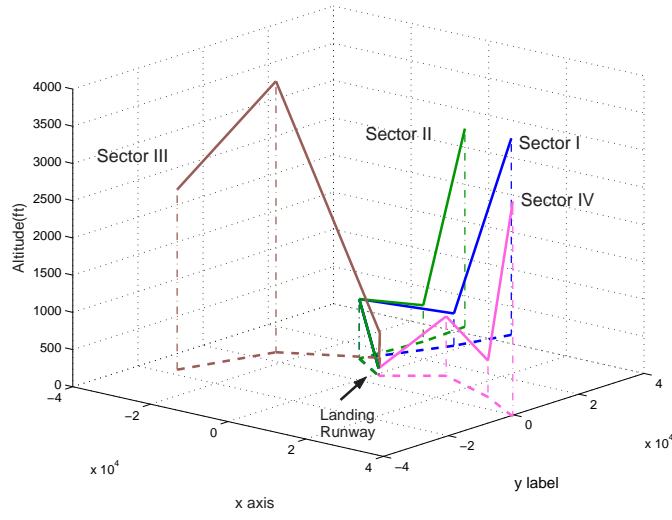
population density distribution, while Figure 4.10(c) and 4.10(d) present the case with population weighting. It is shown that, while the total noise integrated over the entire trajectory is lower for uniform density weighting, inclusion of the population weighting terms yields a result with significantly lower noise over the region around points A and C, the high-population areas.

It is noticed that inclusion of the measurement matrix based on the weights proportional to the population density makes the final noise measurement more accurate. Even with around 400 measurements per cost evaluation, it is still solvable by using the coupled incremental search method (k -ary + Dijkstra).

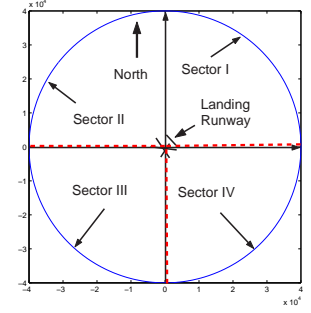
4.3.3 SNI Noise-Optimal Trajectories for Varied Entry Sector

The practical objective of this work is to design SNI approach procedures for RIA. To illustrate the use of our design tool, and to show that SNI approaches are feasible at a major airport (BWI) without altering existing fixed-wing procedures, we conducted a series of optimization runs with traffic obstacles as well as the population and BVI noise models used above.

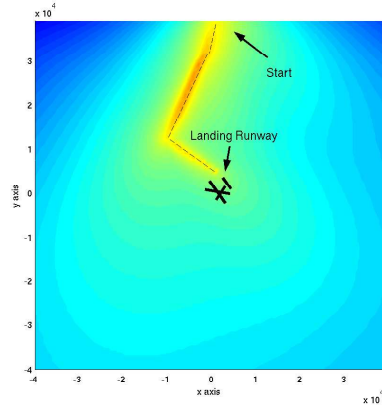
As described previously, east flow traffic at BWI on July 3, 2003 is modeled as a set of impenetrable cylinders and cones for our RIA approach design optimization tool. Rather than finding a single optimal solution, we identified optimal solutions for each approach entry quadrant, facilitating direct approach interception rather than an extended entry that circles far around the BWI terminal area. The entry regions in which the optimizer can place its initial approach fix represent heading



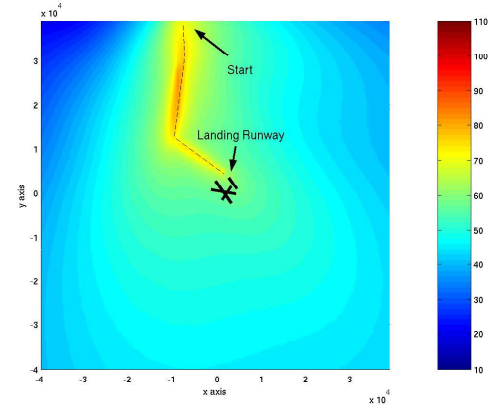
(a) Optimal Trajectories



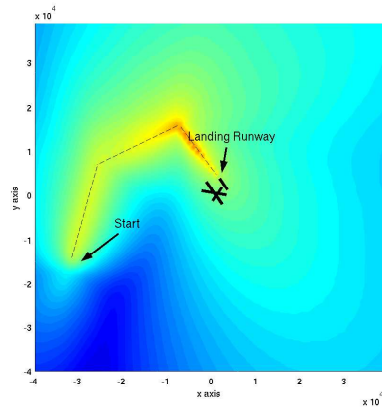
(b) Sector Definition



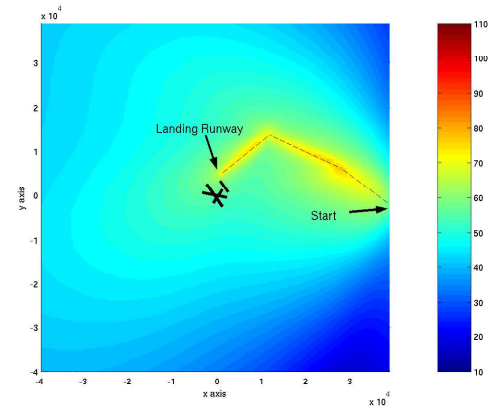
(c) Sector I



(d) Sector II



(e) Sector III



(f) Sector IV

Figure 4.11: Path Shape and BVI SEL Distribution for Traffic from Different Sectors

quadrants (NE,NW,SW,SE) of > 7 -mile radius with altitude above 2,000 ft (shown in Figure 4.11(b)). AH-1 approach trajectories were optimized for each region.

Figure 4.11(a) gives the four locally optimal trajectories for Runway 15L, while Figures 4.11(c)-(f) show the SEL distribution. The final solutions for Sector I and II are similar, utilizing constant descent followed by decelerating climb and descent, a reasonable low-noise solution given the velocity-dependent BVI noise profiles as shown in Figure 2.4. The optimum Sector III trajectory climbs over the aircraft on final approach, then descends to approximately join the final segment from the Sector I/II solutions. For Sector IV, the optimal SNI solution must avoid the traffic departing from Runway 15R, similarly following an alternating climb/descent path to landing. Table 4.2 shows the numerical comparison between these four solutions². First, perhaps the most significant conclusion is that for real RIA and airport models, not only can one SNI approach be found, but in fact SNI approaches of varying quality can be identified for all entry sectors. The Sector II solution is best from a noise perspective and would be the best default procedure identified in this study. The Sector III and IV solutions enable more efficient approaches for traffic entering from the South and also are acceptable from a noise perspective, particularly if used infrequently.

Given the proximity of the Sector I and II solutions, we propose routing traffic in three sectors (red dashed lines in Figure 4.11(b)), favoring the lowest-noise approach from the North. By this means, all RIA traffic entering from the Southwest

²The “0” number of people exposed to noise greater than 60 dBA may not be precise, since our “microphone” matrix may not capture the highest noise area if it is too sparse.

Table 4.2: Comparison of SNI Noise-Optimal Trajectories

Sector	Cost ($10\log J_n$) (dB-PD)	People Exposed to noise > 60 dBA	Fuel Consumption (lb)	Flight Time (s)
I	48.4	981	28.1	444
II	45.6	0	26.1	417
III	53.0	9,301	26.0	440
IV	55.0	19,991	28.4	481

and Southeast would fly the trajectory from Figure 4.11(e) and (f), respectively, while all traffic from the North would utilize the trajectory from Figure 4.11(d).

4.4 Summary

This chapter has presented models and methods to build 3-dimensional noise optimal segmented trajectories for RIA. These trajectories will enable SNI operations since airspace occupied by fixed-wing traffic is strictly excluded from proposed RIA routes. Taking as input a landing site, airport area population and flight track data, and RIA performance constraints and noise model, our discrete optimization tool builds a k -ary tree to represent trajectory segments and uses Dijkstra’s search algorithm to identify the globally-optimal segmented solution. As a practical case study, SNI RIA routes have been computed for an AH-1 rotorcraft on approach to Runway 15L at BWI airport, a viable landing choice for both VTOL and ESTOL vehicles.

Perhaps the most significant result from this study is that true SNI trajectories do exist without alteration to existing fixed-wing traffic patterns. In fact, SNI solutions can be found from all approach directions to BWI 15L, albeit at non-

negligible cost increase when a fixed-wing approach/departure corridor must be avoided. For the AH-1, optimal trajectories are driven to the maximum allowed climb/descent flight path angles to minimize BVI. As might be expected, inclusion of population density guides the optimal flight track away from highly-populated areas and reserves any higher-BVI configuration (e.g., shallow descent) for the final approach segment near the runway threshold where population is sparse.

The segmented trajectory optimization algorithm, BWI population/traffic models, and data interfaces are general for any RIA modeled at any level of fidelity. However, some simplifications were made in our analysis that will require extension in next step, e.g., the most significant assumption is that transitions between trim states do not appreciably affect solution cost.

Chapter 5

Trajectory Planning With Smooth Transitions

In previous chapters, we modeled each SNI trajectory with instantaneous transitions between segments. Recalling the approximate BVI noise characteristics shown in Figure 2.4, we notice there exist noise peak ridges. If a large flight path angle change is allowed between segments, we typically neglect or under-sample the peak noise when the states transfer from one side of the noise peak ridge to the other side. In other words, when the helicopter transitions from a steep climb to a steep descent, the wake must pass briefly through the rotor disk (see Figure 2.2), creating a short-term but high-magnitude BVI. Although the final noise cost is integrated over flight time and the durations of such transitions are normally brief, BVI would be very high and might affect the optimal results. To make the trajectory planning results more accurate, a smooth longitudinal transition must be included. Furthermore, in 3-D cases, we previously assumed instantaneous heading changes since the BVI Q-SAM noise model is only experimentally validated for longitudinal flight. In fact, recent experiments show that Q-SAM can be adjusted for constant turning flight with small to moderate bank angles. We must also consider smooth turning flight and its effects on noise-optimal solution characteristics..

In this chapter, we first present a model of Q-SAM for turning flight. Next, models of smooth transitions - both longitudinal plane and 3-D - are described.

Finally, a comparison with previous SNI solutions and overall analysis are presented.

5.1 Q-SAM Sphere Selection for Turning Flight

Although experimentally validated only for straight flight, we can approximate BVI noise for shallow, steady turns by rotating the Q-SAM noise sphere in accordance with the AH-1 bank angle. Assume the inertial reference $R\{x, y, z\}$. As in our previous cases for longitudinal flight, the median plane (elevation $\theta = 0^\circ$) of the sphere (dashed lines in Figure 5.1) is oriented horizontally and centered at the trajectory point. To get the noise model for turning flight, we simply rotate the sphere about its x -axis by bank angle ϕ with respect to the horizontal axis (see Figure 5.1). For our trajectory design, we presume a fixed bank angle $\phi = 15^\circ$ for all turns.

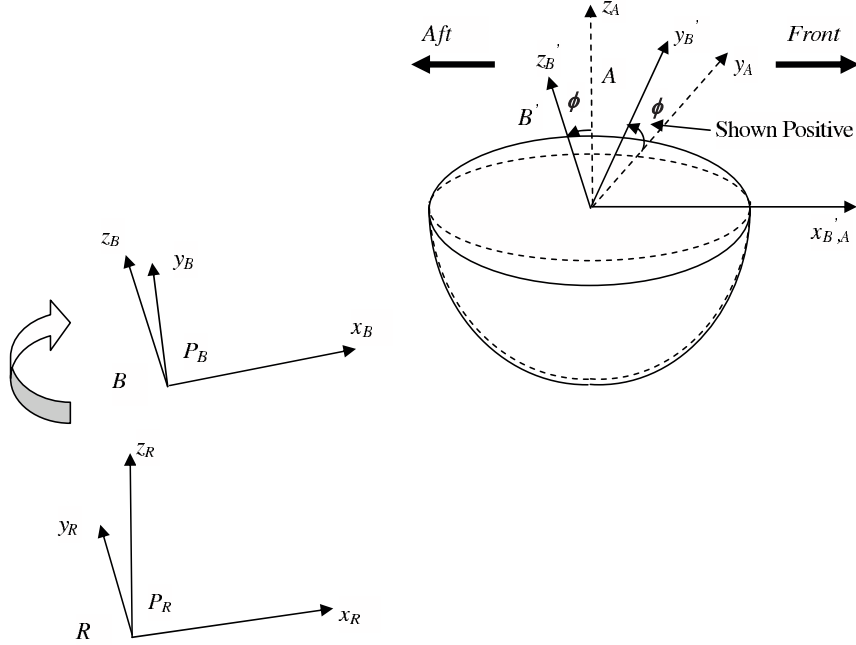


Figure 5.1: Sphere Selection Module for Turning Flight

In the optimization code, we equivalently fix the noise sphere and rotate inertial frame $\{R\}$. Thus when the bank angle ϕ is positive, to get relative elevation and azimuth angle, we rotate inertial frame $\{R\}$ with respect to the x -axis of the body frame by negative angle ϕ . The following shows the convention of the transformation matrix ${}^B_R T$ [48]:

Define the x -direction of the body frame to be $\hat{k} = \{k_x, k_y, k_z\}$, a unit vector. Let the origin of the body frame (coincident with the sphere center) with respect to $\{R\}$ be $P_A = \{p_x, p_y, p_z\}$. Then:

$${}^B_R T = {}^B_{B'} T {}^{B'}_A T {}^A_R T \quad (5.1)$$

$$= \begin{pmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.2)$$

$$\text{where } \begin{cases} r_{11} = k_x k_x (1 - \cos \phi) + \cos \phi, & r_{12} = k_x k_y (1 - \cos \phi) + k_z \sin \phi, \\ r_{13} = k_x k_z (1 - \cos \phi) - k_y \sin \phi, & r_{21} = k_x k_y (1 - \cos \phi) - k_z \sin \phi, \\ r_{22} = k_y k_y (1 - \cos \phi) + \cos \phi, & r_{23} = k_y k_z (1 - \cos \phi) + k_z \sin \phi, \\ r_{31} = k_x k_z (1 - \cos \phi) + k_y \sin \phi, & r_{32} = k_y k_z (1 - \cos \phi) - k_x \sin \phi, \\ r_{33} = k_z k_z (1 - \cos \phi) + \cos \phi. \end{cases}$$

$$\begin{pmatrix} x_B \\ y_B \\ z_B \\ 1 \end{pmatrix} = {}^B_R T \begin{pmatrix} x_R \\ y_R \\ z_R \\ 1 \end{pmatrix} \quad (5.3)$$

After rotation, the new orientation would be calculated through Eqn. 5.3. Then the noise cost at actual location $\{x_R, y_R, z_R\}$ is retrieved by computing the cost at virtual point $\{x_B, y_B, z_B\}$ from the fixed noise sphere shown in Figure 5.1.

5.2 Smooth Transition Models

This section describes the models used for smooth longitudinal transitions and smooth turns. The integration of both models into the 3-D optimization framework is also discussed.

5.2.1 Longitudinal Transitions

Modeling a smooth longitudinal transition is straightforward. To constrain $\dot{\gamma}$ for each non-smooth transition, we utilize a circular arc of radius R with constant velocity as a substitute for the original instantaneous longitudinal transition (see Figure 5.2). Presuming a fixed load factor n during the transition, we will have a fixed centripetal force F . For instance, if $n = 0.1 = F/mg$, we have $a = F/m = 0.1g$, where m denotes the mass of the AH-1, and a is the corresponding acceleration. With a constant velocity V in state P_2 (Figure 5.2), Eqns. 5.4 and 5.5 are obtained

from force balance and the definition of $\dot{\gamma}$, respectively:

$$R = \frac{V^2}{a} = \frac{V^2}{ng} \quad (5.4)$$

$$\dot{\gamma} = \frac{a}{V} = \frac{ng}{V} \quad (5.5)$$

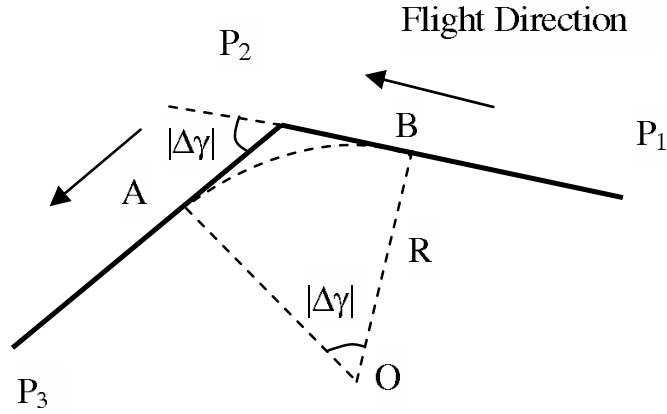


Figure 5.2: Example of Smooth Longitudinal Transition

During optimization, we presume pointwise continuous arcs. After incorporating the smooth transition, the original path $P_1 \rightarrow P_2 \rightarrow P_3$ becomes $P_1 \rightarrow B \rightarrow A \rightarrow P_3$, where we presume arc $B \rightarrow A$ has the same velocity as state P_2 . If we search backward from P_3 to P_1 , presuming P_3 is the landing state with cost $= 0$, P_2 will have the same cost J_{P_2, P_3} as with the instantaneous transition, which means flying from P_2 to landing site P_3 costs J_{P_2, P_3} . Furthermore, P_1 would have the cost given in Eqn. 5.6³, where $J_{B, A}$ denotes the cost of smooth transition BA

³This cost strategy is applied to all calculations with smooth transitions, including smooth pushing/pulling, smooth turning, and combinations of these two.

approximated by a set of small segments.

$$J_{P_1, P_3} = J_{P_2, P_3} - J_{P_2, A} + J_{B, A} + J_{P_1, B} \quad (5.6)$$

5.2.2 Turning Transition

In this work, we presume turns are conducted turning flight at a single trim state with constant bank angle $\phi = 15^\circ$, constant longitudinal path angle γ , constant velocity, and no sideslip. Figure 5.3 shows an example of this model. In the top view, the prescribed path is a circular arc with fixed radius. In the side view (longitudinal plane), this path projects to a straight segment. Presuming constant γ and the constant velocity from state P_2 throughout the turn, and constant γ , we can straightforwardly define turning transitions. As in the smooth longitudinal case, the original path $P_1 \rightarrow P_2 \rightarrow P_3$ becomes $P_1 \rightarrow B \rightarrow A \rightarrow P_3$ with the transition. The cost is calculated in a similar fashion as well.

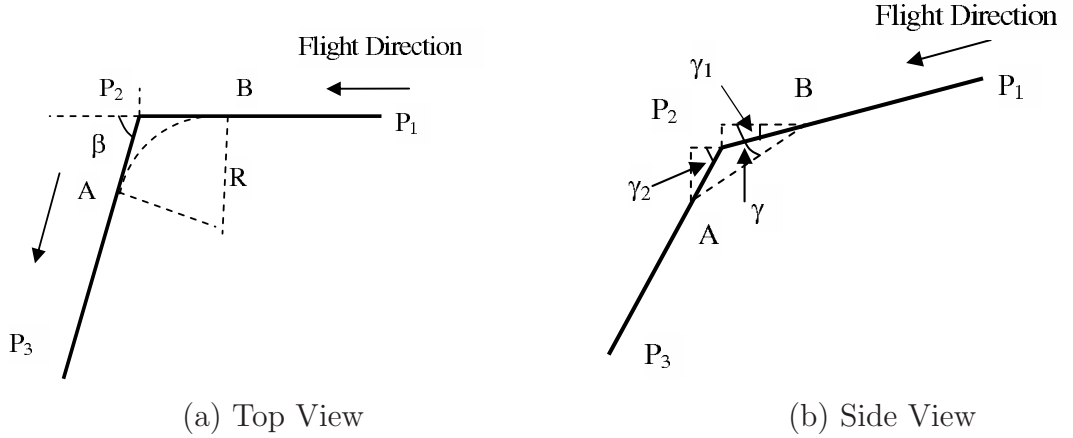


Figure 5.3: Example of Smooth Turning Transition

5.2.3 Combination of Turning and Longitudinal Transitions

In 3-D airspace, there can exist transitions that simultaneous vary γ and heading. For such cases, we generate smooth transitions with the following procedures:

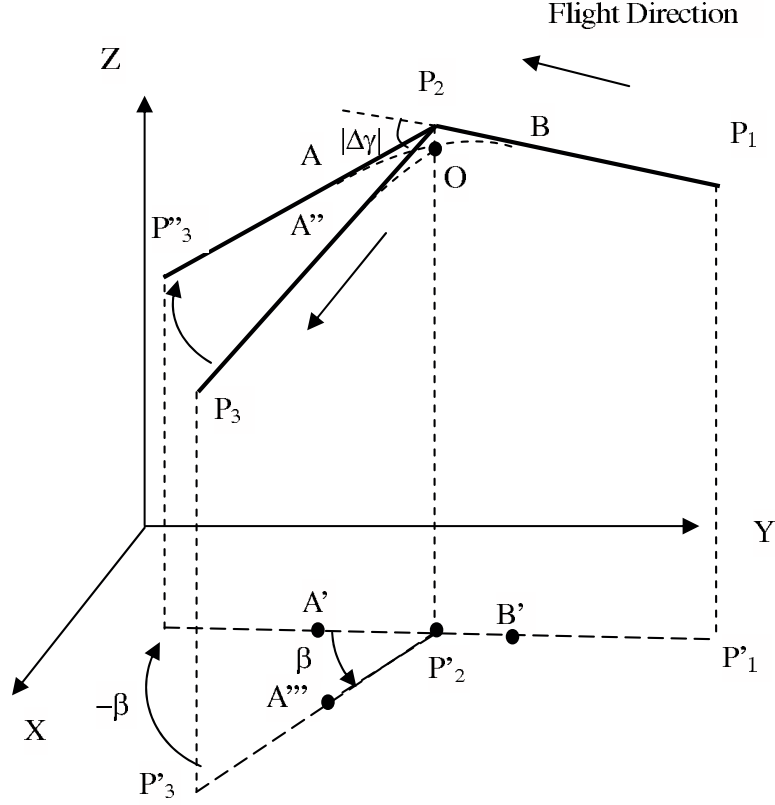


Figure 5.4: Inclusion of Smooth Pulling in Combined Transition

Step 1: Let the projection of sample trajectory $P_1 \rightarrow P_2 \rightarrow P_3$ on the ground plane be $P'_1 \rightarrow P'_2 \rightarrow P'_3$ (see Figure 5.4). Rotating P_3 with respect to axis $P_2P'_2$ by angle $-\beta$, the opposite direction of lateral turning angle β , we generate P''_3 , with P_1 , P_2 , and P''_3 in the same plane.

Step 2: Generate a smooth longitudinal transition arc AOB for $P_1 \rightarrow P_2 \rightarrow P''_3$,

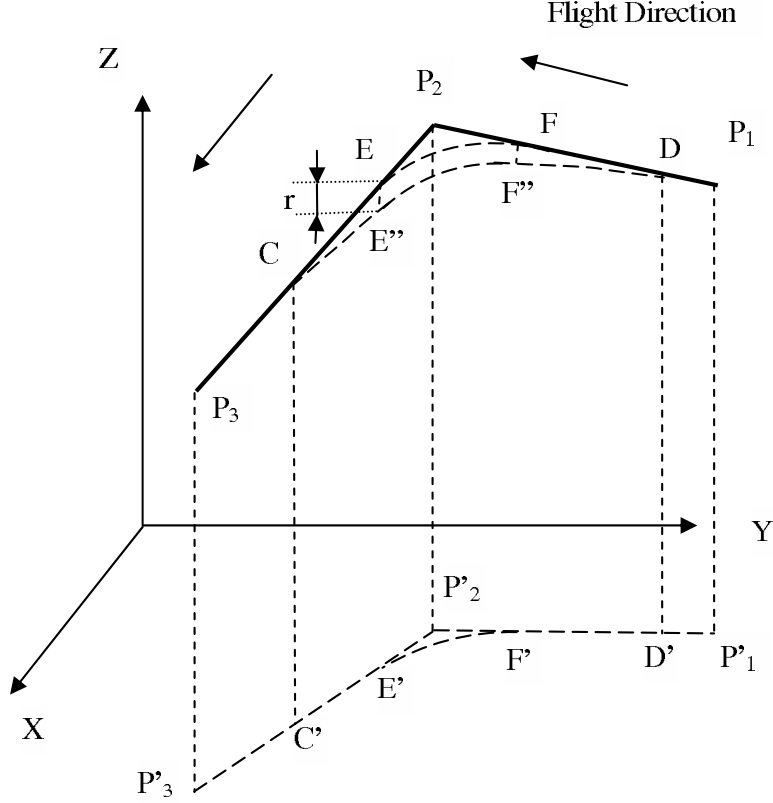


Figure 5.5: Inclusion of Smooth Turning in Combined Transition

where O is the intercept of arc AB with line $P_2P'_2$. The projection of AOB on the ground plane is $A'P'_2B'$.

Step 3: Rotating OA with respect to $P_2P'_2$ by angle β (same direction as lateral turn), we compute OA'' . The projection of OA'' on the ground plane is given by P'_2A''' .

Step 4: Generate smooth turning transition arc EF . The projection of EF on the ground plane is $E'F'$ (see Figure 5.5).

Step 5: Moving EF opposite to the Z -axis direction by distance r (Figure 5.5), we have $E''F''$. Here the distance r is equal to the altitude difference between

points P_2 and O shown in Figure 5.4.

Step 6: Translate OA'' to $E''C$ and OB to $F''D$. The projections on the ground plane are $E'C'$ and $F'D'$, respectively.

Finally, we have combined arc $DFEC$ as the overall smooth transition at point P_2 . The entire transition is presumed to have the constant velocity prescribed for point P_2 , and the original path $P_1 \rightarrow P_2 \rightarrow P_3$ becomes $P_1 \rightarrow D \rightarrow F \rightarrow E \rightarrow C \rightarrow P_3$.

5.3 Simulation Results

In this section, we re-calculate the SNI trajectories to study the effects of including smooth longitudinal and turning transitions. We present the new solutions for arrivals into BWI airport. Finally, the DNL maps for executing the optimal solutions are shown.

5.3.1 Comparison of 2-D Solutions with Instantaneous and Smooth Transitions

To compare with previous 2-dimensional instantaneous transitions, we first generate new solutions with decomposition depth level 6 (corresponding to a 7-segment flight trajectory). Boundary conditions and constraints are all identical to those from Chapter 3. As a baseline, Figure 5.6 recalls the instantaneous transition noise-optimal solution with noise cost 74.10 dBA.

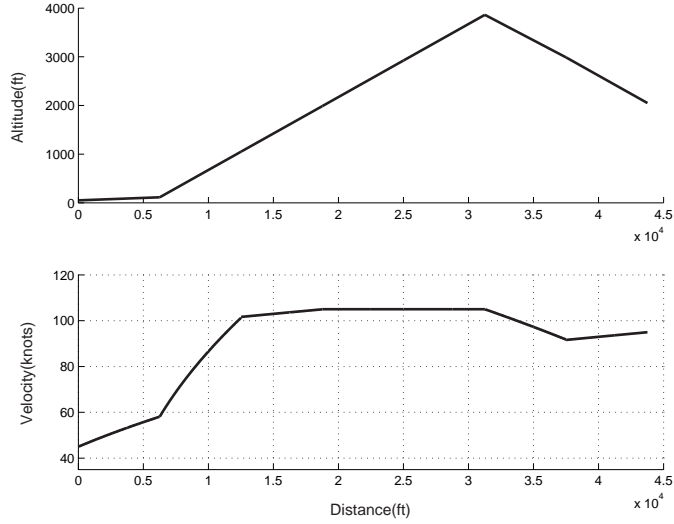


Figure 5.6: Solution with 2-D Instantaneous Transitions (depth = 6, $n = \infty$)

In Figure 5.7, we set the load factor $n = 0.1$, a moderate value with acceptable passenger comfort. The new optimal path and velocity profiles are quite different from the original results. The new solution utilizes fewer transitions, particularly at low altitude, and the transitions are smooth curves with constant velocities. From the noise level perspective, the new cost is 76.51 dBA. The cost 75.40 dBA shown in parentheses neglects the smooth transitions (following the black dash-dot trajectory). Inclusion of the smooth longitudinal transitions increases the final costs since it includes the high BVI noise values resulting from the transition through the maximum noise ridge. Although the solution presented in Figure 5.6 appears to have lower cost, it introduces significantly more noise in practice (Costs become 77.70 dBA when $n = 0.1$).

To analyze the effect of load factors on the final solutions, we varied the load factors from low to high values and re-ran the 2-D optimization process. When $n = 0.05$, the basic path in Figure 5.8 is the same as to the solution with $n = 0.1$

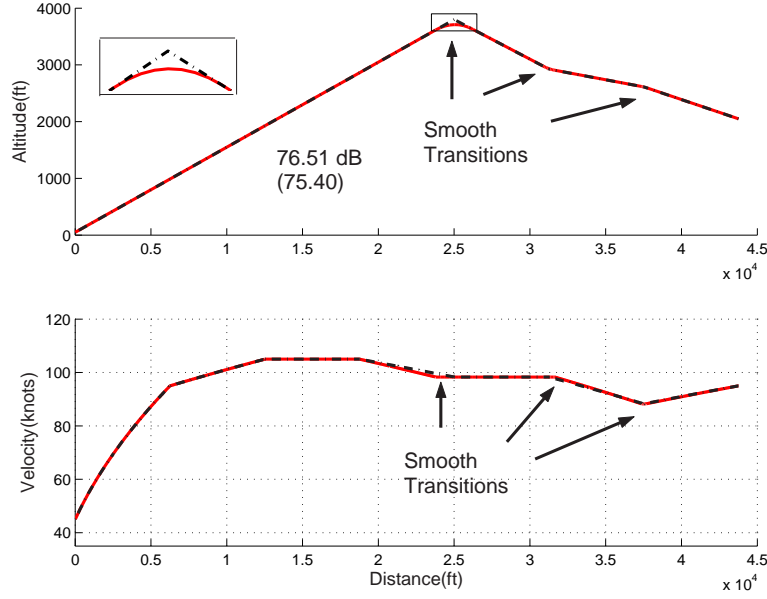


Figure 5.7: Solution with Smooth Longitudinal Transition (depth = 6, $n = 0.1$)

and dynamic profiles are similar, except for slight transition shape difference and increased noise cost. In the case of Figure 5.9, where the load factor is 0.5, we find that the optimizer uses similar path and velocity profiles to the baseline (since $n = 0.1$ will cause a much longer transition time, the cost is 75.10 dBA with $n = 0.5$). That means if the load factor is allowed to be large ($n \geq 0.5$), the instantaneous transition model is a reasonable approximation that slightly underestimates noise cost. For passenger-carrying operations, $n = 0.1$ is considered a reasonable limit. This 2-D analysis suggests inclusion of the transition model will in fact appreciably affect results.

When the decomposition depth level is increased to 7 (corresponding to a 13-segment trajectory), as shown in Figure 5.10, the final solution (red line) also changes significantly compared with the baseline (solid black line). The final solution contains fewer large-magnitude transitions. Including smooth transitions in this

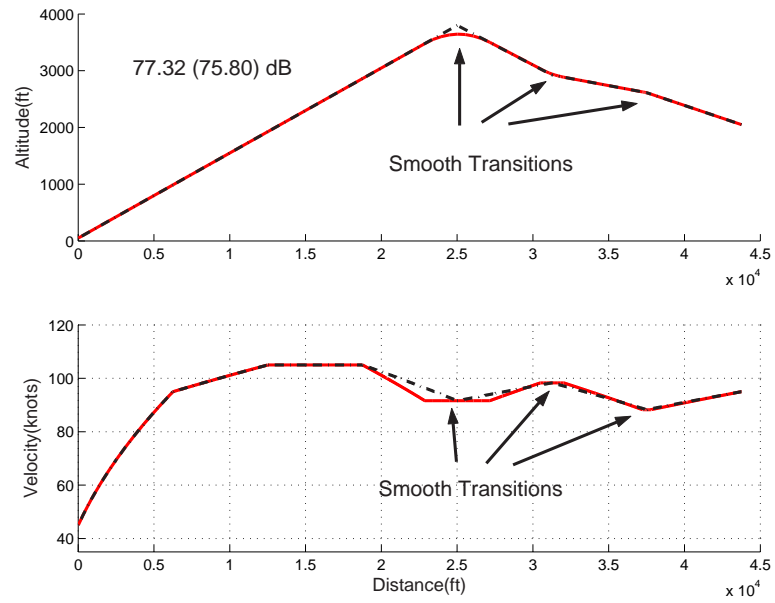


Figure 5.8: Solution with Smooth Longitudinal Transition (depth = 6, $n = 0.05$)

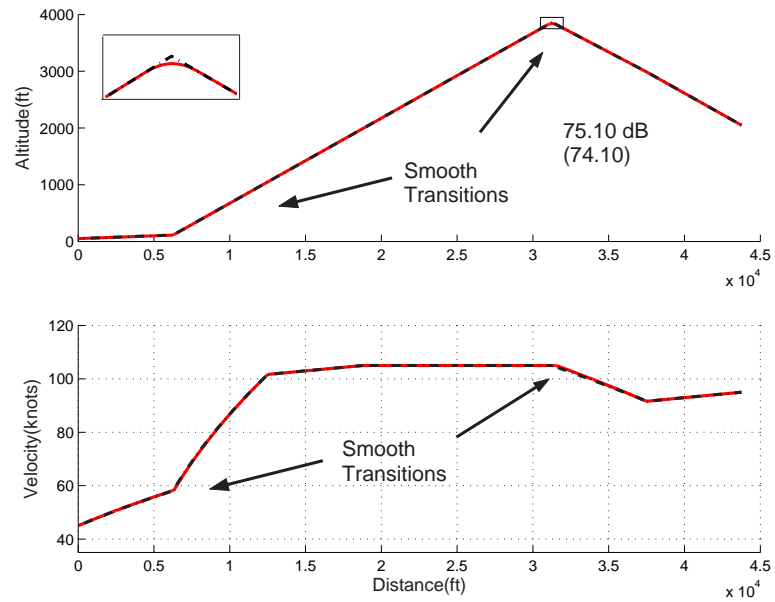


Figure 5.9: Solution with Smooth Longitudinal Transition (depth = 6, $n = 0.5$)

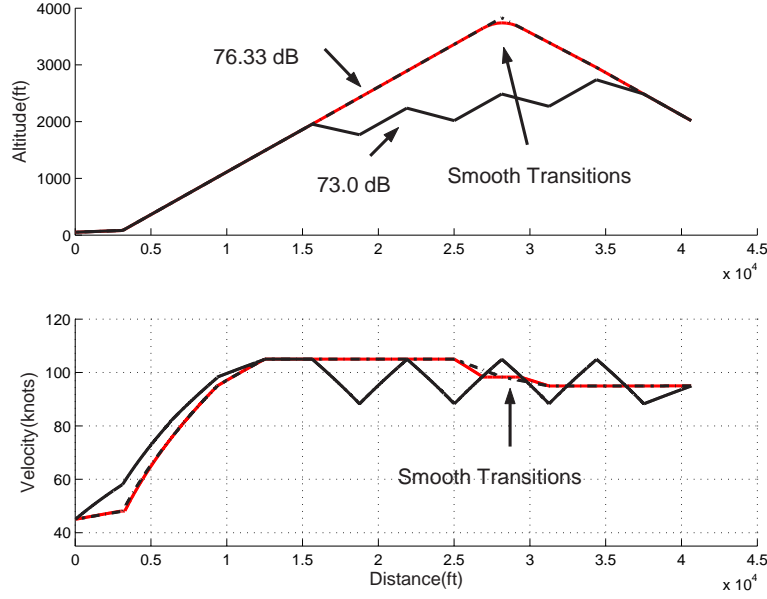


Figure 5.10: Solution with Smooth Longitudinal Transition (depth = 7, $n = 0.1$)

case eliminates the unrealistic zig-zag path shape caused by incomplete trajectory modeling, a particularly important feature with high iterative deepening search depth levels.

5.3.2 Comparison with 3-D Instantaneous Transitions

With the model of smooth 3-D transitions described above, we repeated the 3-D optimization for Sector II at BWI (Chapter 4). Figures 5.11 - 5.13 show the optimal solutions with load factors $n = 0.5, 0.1$, and 0.05 , respectively. For illustrative purposes, one of the two smooth transitions in each trajectory are amplified in the figures. The optimized trajectory profiles in these figures are still the same as with $n = \infty$ except for the smooth transitions. But the costs are increased from 45.6 for $n = \infty$ dBA to 46.44 dBA, 47.32 dBA, and 47.80 dBA, respectively. Compared with the baseline, including smooth transitions means capturing more high noise

values around noise peaks; consequentially, the final costs will be increased.

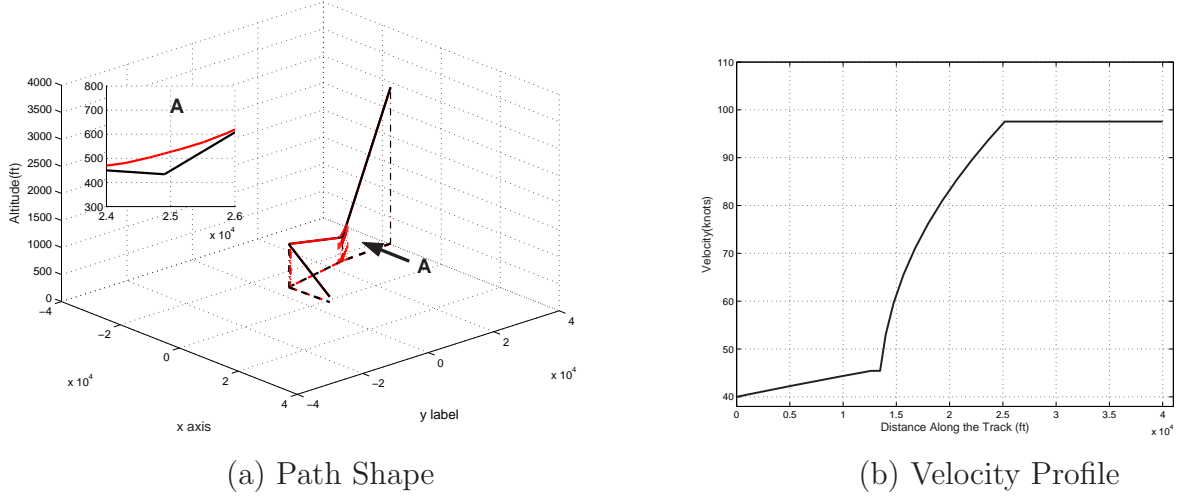


Figure 5.11: 3D Optimal Solution with Smooth Transition ($n = 0.05$)

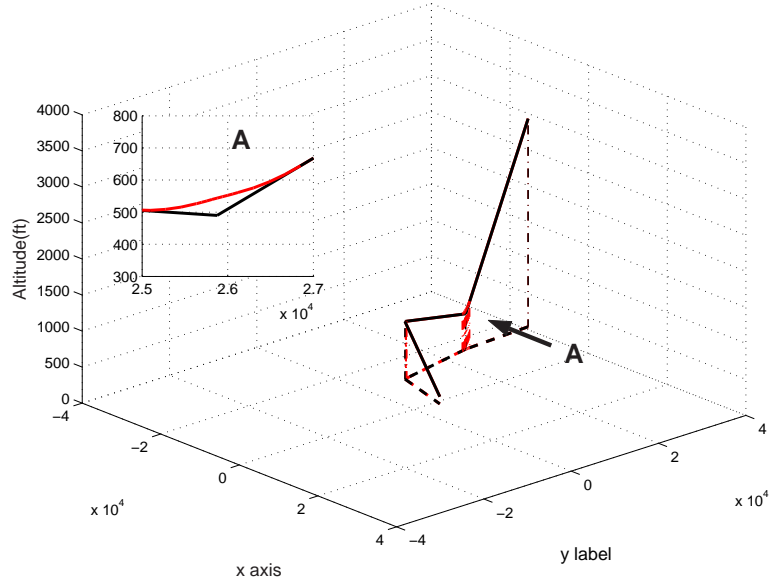


Figure 5.12: 3D Optimal Solution with Smooth Transition ($n = 0.1$)

5.3.3 SNI Optimal Solutions with Varied Arrival Sectors

To demonstrate the effects of smooth transitions on all SNI solutions identified for BWI airport, we show a full set of noise-optimal trajectories for the four

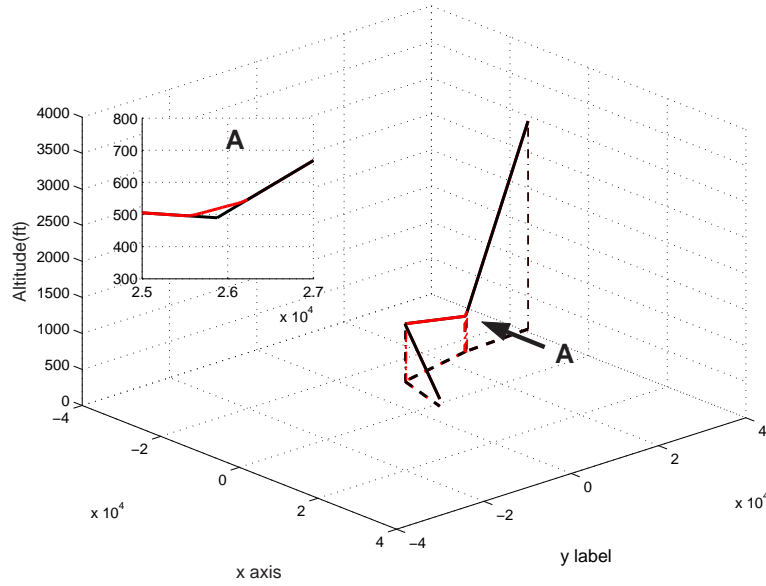
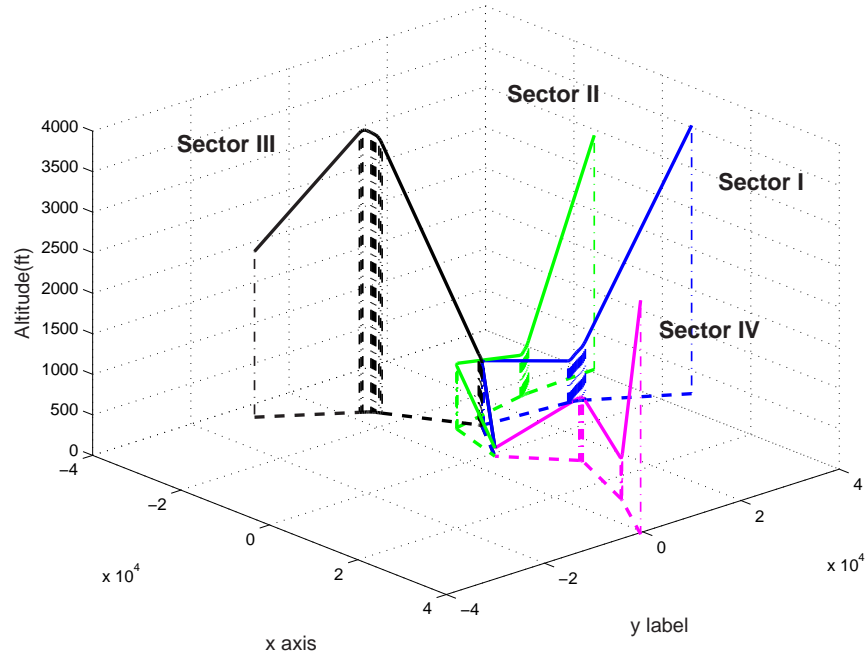


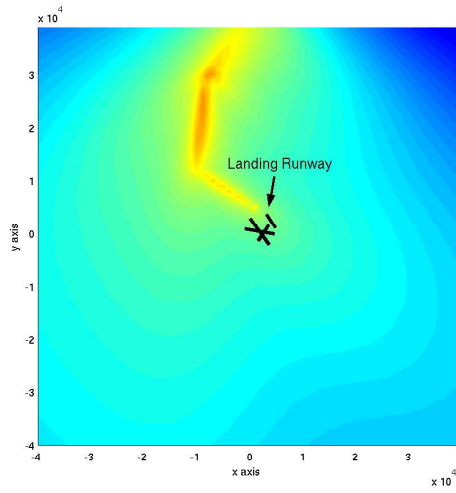
Figure 5.13: 3D Optimal Solution with Smooth Transition ($n = 0.5$)

arrival sectors with load factor $n = 0.1$. The path profiles are presented in Figure 5.14(a). Again, ground noise distributions in SEL are shown in Figure 5.14(b-e). The path shapes are similar to the baseline solutions except for the inclusion of smooth curves during transitions. The SEL maps have changed significantly due to inclusion of smooth transitions, e.g. in Figure 5.14(d) the last transition of the trajectory generates much more noise than for the baseline case.

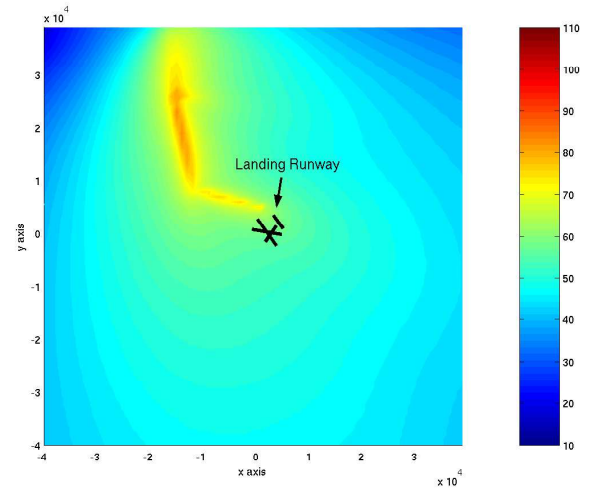
Table 5.1 lists the costs corresponding to these four solutions. As a comparison, we present the values in parentheses for the instantaneous transition baseline results. First, the number of people exposed to noise levels greater than 60 dBA is higher than for the baseline ($n = \infty$) case. Table 5.1 also shows noise exposure above 65 dBA and 70 dBA. Generally, higher costs indicate more people exposed to noise, except that in the case of people exposed to noise greater than 65 dBA, the Sector II approach actually exposed more people to noise > 65 dBA than does the Sector



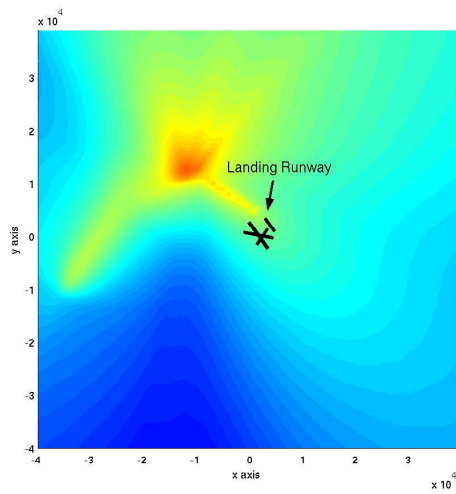
(a) Optimal Trajectories



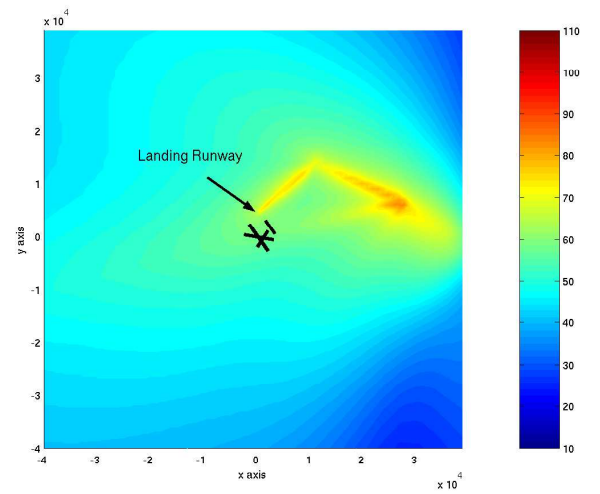
(b) Sector I



(c) Sector II



(d) Sector III



(e) Sector IV

Figure 5.14: Path Shape and BVI SEL Distribution for Traffic from Different Sectors

I approach.

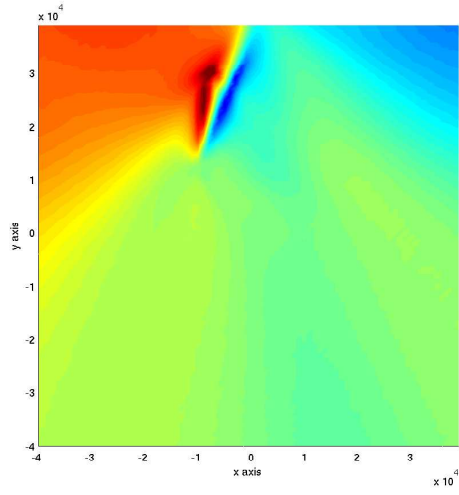
Table 5.1: Comparison of SNI Noise-Optimal Trajectories with Smooth Transitions

Sector	Cost ($10\log J_n$) (dB-PD)	People Exposed to Noise		
		> 60 dBA	> 65 dBA	> 70 dBA
I	52.2(48.4)	8,736(981)	0	0
II	47.3(45.6)	5,490(0)	55	0
III	61.1(53.0)	95,422(9,301)	28,856	7,581
IV	58.6(55.0)	38,195(19,991)	12,472	3,522

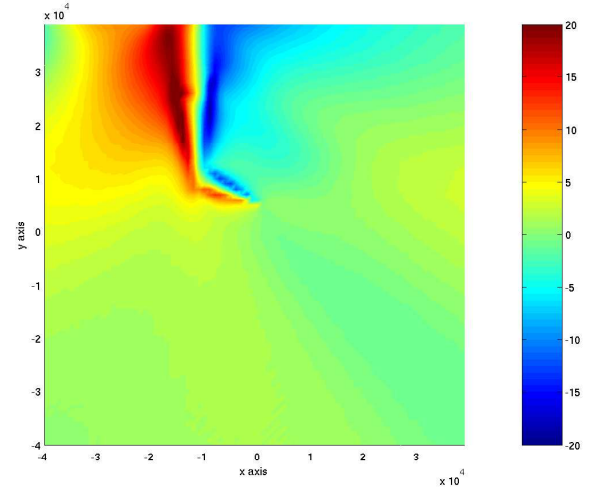
The differences in noise cost between instantaneous and smooth transition solutions are presented in Figure 5.15(a-d). Warmer colors (e.g. red) denote where the new solution generated higher noise than the baseline solution, while the cooler colors (e.g. blue) represent cases where the new solution was less noisy. In Figure 5.15(a) and (b), since fixed-wing corridors do not significantly constrain the 3-D space for Sector I and II, the optimizer utilizes turning transitions to expose the populous area (the top-right corner) to less noise. In Figure 5.15(c) and (d), because of the existing fixed-wing traffic routes, there is no obvious way to utilize turning transitions to direct noise away from the population centers.

5.3.4 DNL Distribution Map

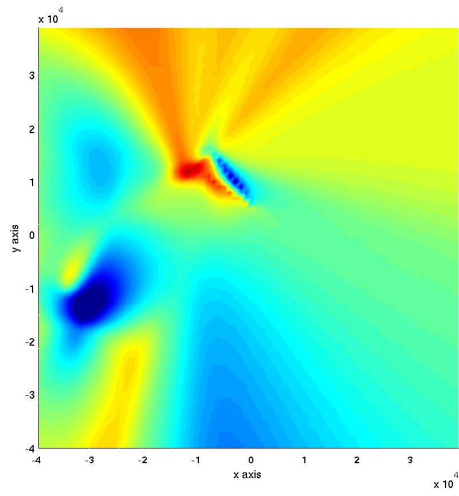
In practice, airport planners measure noise annoyance with an integrated Day-Night Average Sound Level descriptor (DNL)[45], which is a composite noise index derived from a 24-hour average noise level with a weighting factor of 10 decibels applied to noise events occurring at night (10 pm - 7 am). Thus, our SEL distribution can be converted to DNL through:



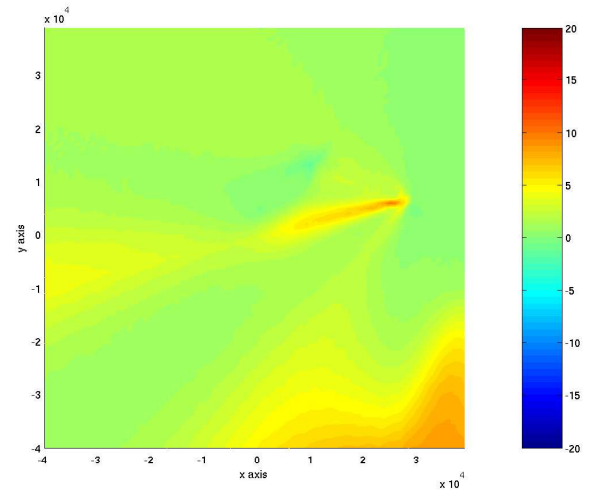
(a) Sector I



(b) Sector II



(c) Sector III



(d) Sector IV

Figure 5.15: BVI SEL Difference Map with Varied Arrival Sectors

$$DNL = 10 \times \log(10^{\frac{SEL_{pf}}{10}} \times \frac{flightsperday}{duration}) \quad (5.7)$$

where SEL_{pf} is SEL per flight (integrated over the full SNI approach). Since only daytime operations are considered, the *duration* is set as 15 hours ⁴. The DNL distributions for the Sector II AH-1 solution are shown in Figures 5.16 and 5.17. Totals of 100 and 300 flights per day represent 6.6 flights/hour and 20 flights/hour, respectively. This operational assumption is the worst-case scenario, since it represents exact overflight of the same areas, while any difference of trajectories would make the DNL lower. As indicated in [45], 65 DNL or less is considered an acceptable level for residences, schools, and hospitals, while 70 DNL or less is acceptable for hotels, playgrounds, etc. Even with a busy SNI approach day of 300 flights (20/hour), the BVI-related DNL distributions for the AH-1 on approach to BWI are still low compared with normal operations [45] into BWI, where many areas are exposed to 75 DNL.

Based on the experimental results presented in [49] and [50], it is possible to scale the predicted AH-1 noise to the "actual" AH-1 noise by subtracting approximately 15 dBA from the noise database we are using. The differences are primarily due to the lack of unsteady aerodynamics modeling and non-compact chord modeling [50]. To transfer obtained DNL maps to a typical commercial RIA vehicle, which is expected to weigh approximately 50,000 lb, we assume that BVI noise scales only with vehicle gross weight. Since AH-1 weighs about 10,000 lb, the additional noise

⁴SNI operations would likely not be conducted at night since throughput requirements are typically much lower.

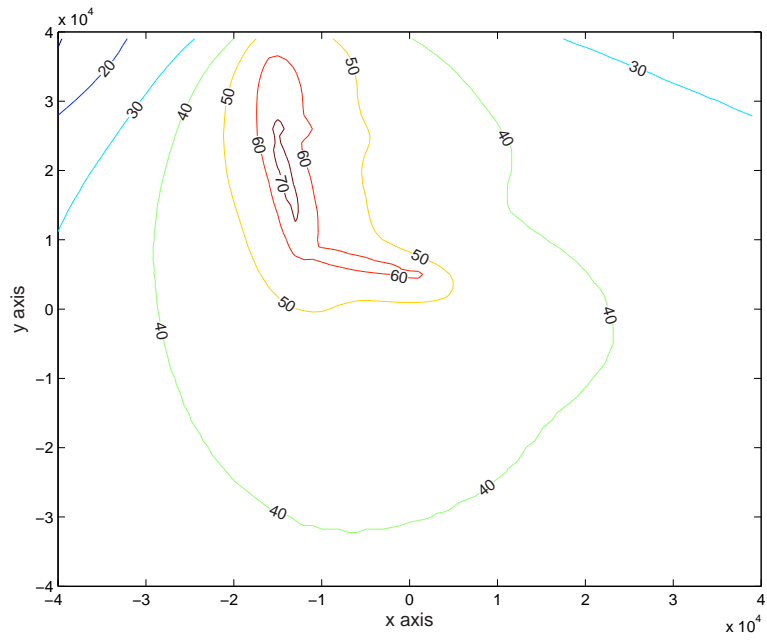


Figure 5.16: DNL Distributions with Sector II Approach(100 Flights/Day)

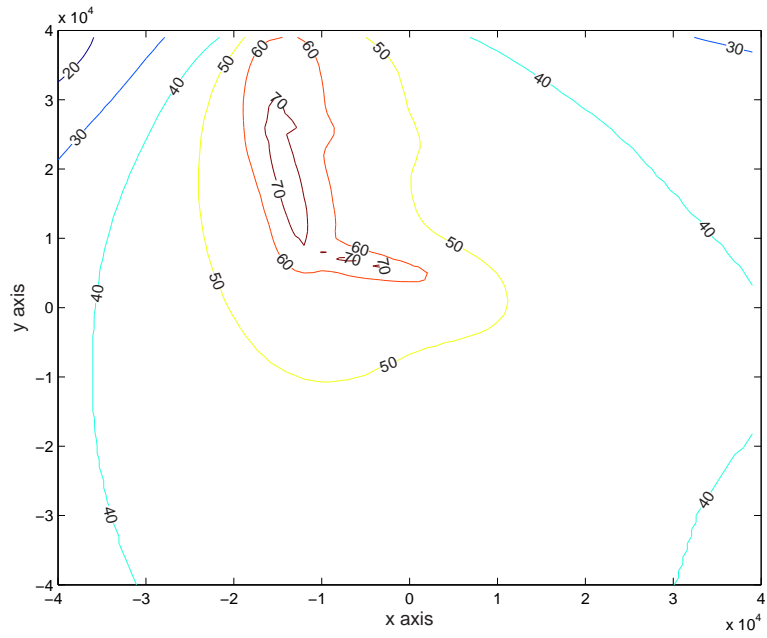


Figure 5.17: DNL Distributions with Sector II Approach(300 Flights/Day)

contribution is about $20\log_{10}(50,000/10,000) = 14$ dB. Therefore, this larger-scale RIA vehicle is expected to generate DNL maps comparable to those presented in Figures 5.16 and 5.17 .

5.4 Summary

This chapter augments previous analysis with smooth transitions between trim segments and presents updated results for RIA SNI noise-optimal approaches to BWI. From the simulation results, it is noted that although the final noise cost is integrated over flight time and normally the durations of such transitions are quite brief, high BVI noise during the transitions still appreciably increases the final cost. Roughly speaking, for a trajectory with a high number of segments, the inclusion of smooth longitudinal transitions prevents the optimal trajectory from degenerating to a bang-bang flight-path-angle solution. With few approach segments, the final trajectory design does not typically vary significantly, but noise costs increase to more realistic levels. In 3-dimensional cases, turning flight segments enable the optimizer to position turns such that they direct noise away from populous areas, except when the solution space is highly constrained by fixed-wing traffic corridors.

Realistic ground noise maps are presented for the BWI solutions for four different entry sectors. The DNL maps are also presented indicating that even with dense operations along the same optimal SNI trajectory, DNL limits are still met and are lower than statistical noise generated by fixed-wing operations. Scaling to a larger rotorcraft would not change the DNL maps significantly. Certainly, including other

noise sources will increase DNL for the SNI operations, but these results validate the SNI concept both from route existence and DNL acceptability perspectives.

Chapter 6

Real-time Trajectory Planning

Thusfar, we have used an incremental search method which couples cell-decomposition (2-D) / k -ary tree (3-D) methods with Dijkstra's search algorithm. This methodology provides an effective way to find the global optima given a discrete search space, but it still requires substantial search that becomes prohibitively complex with increasing number of segments and/or resolution of the design parameters. With the inclusion of the smooth transition models, this method requires around 1 hour to obtain a 3-D solution even with rough discretization, e.g. resolution = 8 which represents 8 values per design variable for a 3-segment trajectory. To improve optimization efficiency and analyze the effects of parameter discretization, we investigated two promising randomized optimization methods: the genetic algorithm (GA) and simulated annealing (SA), since they typically perform well for problems with no gradient information and large search spaces.

With all the RIA and airport related models from Chapter 4, this chapter begins by investigating the genetic algorithm and simulated annealing algorithms applied to our discrete optimization problem. Next, with our previous incremental search strategy as a baseline algorithm, a comparison among algorithms is presented to identify the most promising candidate for real-time implementation. Finally, the effects of discretization of the 3D search space is examined specifically with

adaptive simulated annealing, the top performing algorithm. Note that we divide the discretization into two categories: resolution of design variables and number of trajectory segments. In the simulations, both trajectory models with and without smooth transitions will be examined.

6.1 Algorithms

In this section, we again study SNI RIA NAP trajectory optimization with performance and noise models of the AH-1 rotorcraft, and traffic and population models at BWI airport. Although specific RIA and airport models are presented, this problem includes features relevant to all terminal area trajectory optimization problems (e.g. obstacle avoidance, empirical cost function as a “black box”, no derivative information, hard flight envelope limits, and GIS factors like population density). Thus, the comparative study of optimization approaches should provide complexity results general for SNI trajectory design for any vehicle and any airport.

Recall that SNI final approach trajectory optimization is defined as a two-point boundary value problem in three dimensional airspace. Optimized noise abatement procedures are described by a sequence of segments, each with constant velocity or acceleration. In this section, we first show the solution structure, then we describe the GA and SA algorithms applied to our problem.

6.1.1 Decomposition Strategy

In previous chapters, we utilized a k -ary tree to construct the search space based on control variables γ , \dot{V} , β , and distance step Δd in the lateral (xy) plane. As before, we set the number of segments (directly related to Δd) in advance, thus the optimization would be based on variables γ , \dot{V} , and β . Figure 6.1(a) shows the solution structure for an example 3-segment trajectory. Any general n -segment solution will consist of n nodes (note that landing state has been specified). Each node contains three parameters: longitudinal path angle γ , velocity of the RIA V , and lateral path angle β . Thus, given a landing orientation and position, segment i will be identified by the landing parameters plus an acceleration value. The sample optimization parameters (in the case of a 3-segment trajectory) are shown in Figure 6.1(b). For convenience, parameters γ , V , and β are discretized with the same resolution, although it would be possible to have different resolutions for each or to extend to continuous space for simulated annealing.

6.1.2 Dijkstra's Algorithm

Dijkstra's algorithm works on the principle that the least costly path from the source has to be the path traversed by the first visitor from the source to the destination, given its nature to feasibly explore the nodes in order of increasing cost. An intuitive way to think about this is the "explorer" model: starting from the source, we can deploy explorers each traveling at a constant speed and crossing each edge in time proportional to the weight of the edge (i.e. the cost to travel

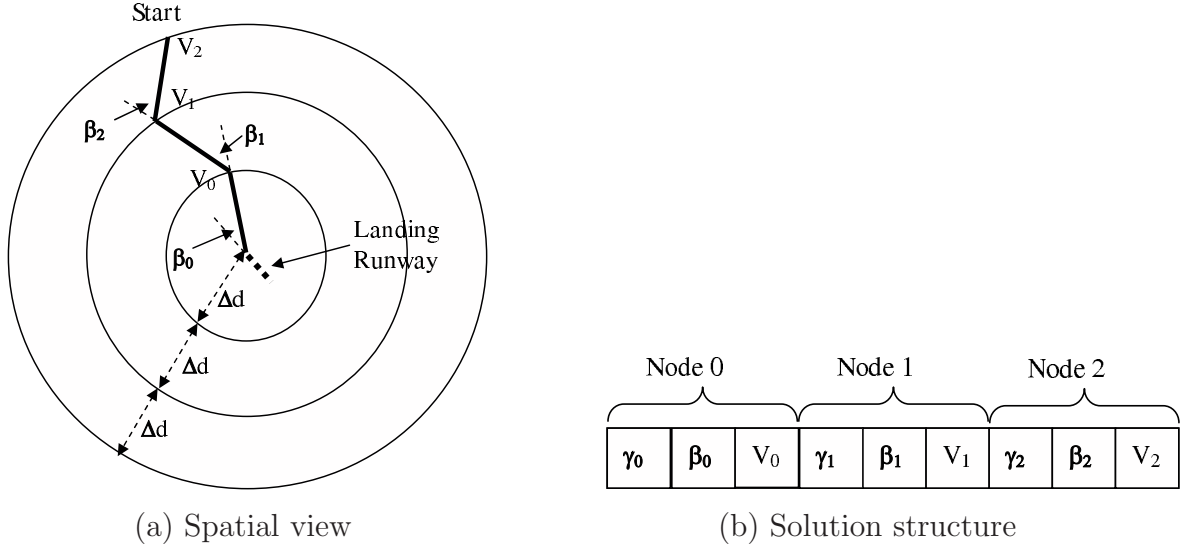


Figure 6.1: Solution Structure

between the two nodes) being traversed. Whenever an explorer first reaches the destination, it records the path it took to get to that vertex, and since this was the least cost of all nodes the explorer is known to have taken the optimal path to reach the vertex.

There are two issues to note. First, every weight/cost is initialized to infinity, except the one from the source itself. Second, in this work, cost evaluations take the majority of the computation time, thus a significant amount of time would be required to pre-build the whole network with a weight for each link. This computation could even be infeasible for a 3D airspace. Thus, in Chapters 4 - 5, we incrementally built up the search space, forming a k -ary tree. The “build” process is synchronous with the search process in Dijkstra’s algorithm. The build and explore processes halt when the destination is first visited.

6.1.3 Genetic Algorithm

The genetic algorithm (GA) [51] is a stochastic process whose search method models two natural phenomena: genetic inheritance and Darwinian evolution [51]. It first creates a population of potential solutions, each solution is called a “chromosome” represented by a binary string of length $m = \sum_{i=1}^k m_i$ for a solution with k design parameters (with $\gamma, \beta, V, k = 9$ for a 3-segment trajectory). The first m_i bits of the string correspond to the first design parameter called the first “gene”, the next group of m_i bits will map into the second design parameter (gene), and so on. In each generation, the population of chromosome is evaluated with a cost function. The new population is then selected with respect to a probability distribution based on fitness values (costs). Finally, the chromosomes are altered in the new population by mutation and crossover operators. The theoretical principle for the GA is that a genetic algorithm seeks near-optimal performance through the juxtaposition of short, low-order, high-performance schemata, called building blocks. In this work, for the selection process, a roulette wheel with slots sized according to fitness is used. Based on empirical work, the crossover probability is set to 0.8 and mutation probability is set to 0.05.

6.1.4 Simulated Annealing

Simulated annealing (SA) is a stochastic strategy inspired by metallurgical annealing. While most optimization techniques are likely to get stuck in local minima, SA is one of the most efficient methods developed to escape from local minima

by allowing tunneling, variable sampling, and hill-climbing. SA was first proposed by Metropolis [52] for simulating a collection of molecules, then Kirkpatrick [54] formally adopted it as an optimization technique. Their work can be categorized as Boltzmann Simulated Annealing [53] (BSA), with state generation governed by a Gaussian distribution. For faster implementation, Szu [55] developed Fast Simulated Annealing (FSA) by using a Cauchy distribution as a state-generation function. The theoretical basis for SA to reach the global optima is that the stochastic state-generation process, which forms Markov chains, should be ergodic [56].

The method used in our work is based on Adaptive Simulated Annealing (ASA) or Very Fast Simulated Annealing (VFSA) introduced by Ingber [57]. In this work, reannealing has not been included and the state generation has been slightly changed in terms of the empirical values. The algorithm is described as:

Suppose x_k^i is the i th dimension parameter generated at annealing time k with range $x_k^i \in [A_i, B_i]$. For 3-segment SNI RIA trajectory optimization $\vec{x} = \{\gamma_0, \beta_0, V_0, \gamma_1, \beta_1, V_1, \gamma_2, \beta_2, V_2\}$ and dimension $D = 9$.

(1) State-generation for D parameters $x = x^i; i = [1, \dots, D]$ with $x_{k+1}^i \in [A_i, B_i]$ is given by:

$$x_{k+1}^i = x_k^i + y^i \quad (6.1)$$

where y^i is a random variable generated by probability density function:

$$g_T^i(y^i) = \frac{1}{2(|y^i| + T_i) \ln(1 + 1/T_i)} \quad (6.2)$$

Here T_i is the “annealing temperature” for the i th design variable.

(2) Acceptance probability density function $h_T^i(x_{k+1}^i)$ when cost increases relative to the previous value is computed by:

$$h_T^i(x_{k+1}^i) = \frac{1}{1 + \exp(\frac{E_{k+1} - E_k}{T})} \quad (6.3)$$

where E represents the “energy” difference between the present and previous values of the energies (considered here as cost functions) appropriate to the physical problem.

(3) The Cooling schedule in annealing time step k is specified as:

$$T_i(k) = T_{0i} \exp(-c_i k^{\frac{1}{D}}) \quad (6.4)$$

where c_i is constant.

In practice, Simulated Quenching (SQ) [58] makes SA much faster although state-generation may not occur infinitely often in time (i.o.t) since the ergodicity of the stochastic process cannot be guaranteed. In our case, SQ is empirically verified to be faster to attain global optima. Quenching factor Q is set to the value of the design vector dimension ($Q = D = 9$). Thus, the cooling schedule becomes:

$$T_i(k) = T_{0i} \exp(-c_i k^{\frac{Q}{D}}) \quad (6.5)$$

6.2 Optimization Procedure

To generate a full set of results, trajectory models are considered both with instantaneous and smooth transitions. In former cases, the solution structures described previously could be directly used. With smooth transitions, as in Chapter 5, many small segments are inserted between the original segments to denote the piece-wise continuous transition curves. But the design variables are still the same, $3 \times N$ for an N -segment trajectory.

Based on the AH-1 and BWI models, we compared the GA and ASA optimization algorithms using our Dijkstra search method as a baseline. In the following cases, the landing site is again fixed at BWI Runway 15L and the entry region is specified as Sector II, the Northwest approach quadrant. Here the final velocity is 40 knots, and the initial velocity is any value above 95 knots. The trajectory spans the area within 7 miles (approximately 40,000 ft) from BWI and to a ceiling of 4,000 ft , with minimum altitude constraint of 100 ft . All cases are run on a 2.8GHZ Linux platform.

6.3 Optimization Algorithm Comparison

With our incremental search strategy, we built each search space by discretizing all design variables over their specified ranges and resolutions (i.e. the number of discrete values per design variable). Since the GA needs to discretize the continuous space, for fair comparison, SA will also be applied to the same discrete search space. In this section, six test cases are constructed with different resolutions: three with

instantaneous transitions and the others with smooth transitions. Case I has a resolution of 8, indicating variables path angle γ , lateral angle β , and velocity V are discretized into 8 discrete values evenly separated over their allowed ranges. Cases II and III have resolutions 16 and 32, respectively. Accordingly, the GA has gene length 3, 4 and 5 for Cases I, II and III respectively. Cases I', II', and III' correspondingly incorporate the smooth transitions. Due to the emphasis on algorithm speed, a strict upper bound on runtime of 500 or 3600 seconds serves as an additional stop criteria for the algorithms.

SA requires an initial solution to start the search. Theoretically, the GA does not need an initial guess since its initial population is generated randomly. However, for this heavily constrained problem, empirical evidence has indicated that, frequently, none of the initial chromosomes is feasible when randomly generated even with a population size greater than 10,000. Thus the same initial guess is specified for SA and inserted into the initial population in GA. Additionally, both GA and SA need a random number generator. For each realization, we use a pseudo-random generator with different seeds for each execution. The initial values are shown in Table 6.1. Cost and computational complexity results for the baseline algorithm are summarized in Table 6.2⁵.

Because SA and GA are statistical methods, for each case 20 realizations were executed with upper time bounds set to 500 seconds for Cases I, II, and III and 3600 seconds for Cases I', II', and III'. Figure 6.2 shows the probability that

⁵We did not complete Case III with instantaneous transitions and resolution = 32, and Cases II' and III' with smooth transitions and resolutions = 16, 32, since the baseline algorithm fails to find the optimal solution within feasible computational time due to the high computational complexity.

Table 6.1: Initial Values for Different Segment Trajectories

Seg.	γ_0	β_0	V_0	γ_1	β_1	V_1	γ_2	β_2	V_2	γ_3	β_3	V_3	γ_4	β_4	V_4
2	-3.5	0.0	45	-3.5	0.0	95									
3	-3.5	0.0	45	-3.5	0.0	70	-3.5	0.0	95						
4	-3.5	0.0	45	-3.5	0.0	60	-3.5	0.0	75	-3.5	0.0	95			
5	-3.5	0.0	45	-3.5	0.0	55	-3.5	0.0	70	-3.5	0.0	85	-3.5	0.0	95

Table 6.2: Optimal Solutions using Baseline (Dijkstra's) Algorithm

Case	Cost (dB-PD)	Computation Time (s)
I	49.80	157
II	43.18	24529 (around 7 hrs)
I'	50.50	2849

each algorithm will identify the global optima for each data run. The baseline Dijkstra's algorithm is guaranteed to attain global optima, thus it is not shown here. Figure 6.2(a) presents the comparison among trajectories with instantaneous transitions, while Figure 6.2(b) represents cases that model smooth transitions. This comparison indicates that SA performs better than the GA for both scenarios, since SA is most likely to attain the globally optimal solution. Empirical works show that

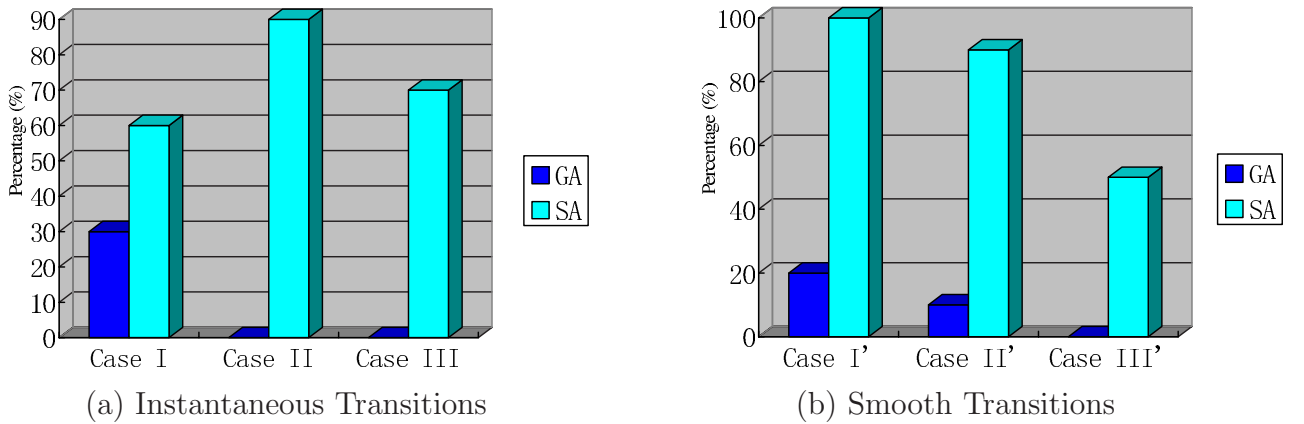


Figure 6.2: Percentage of Attaining Global Optima

at the upper time bound the GA is typically far from the global optima, especially for large search space sizes. Additionally, initial guesses do not affect the performance of both GA and SA substantially. GA requires feasible initial guesses to find even a suboptimal but feasible solution.

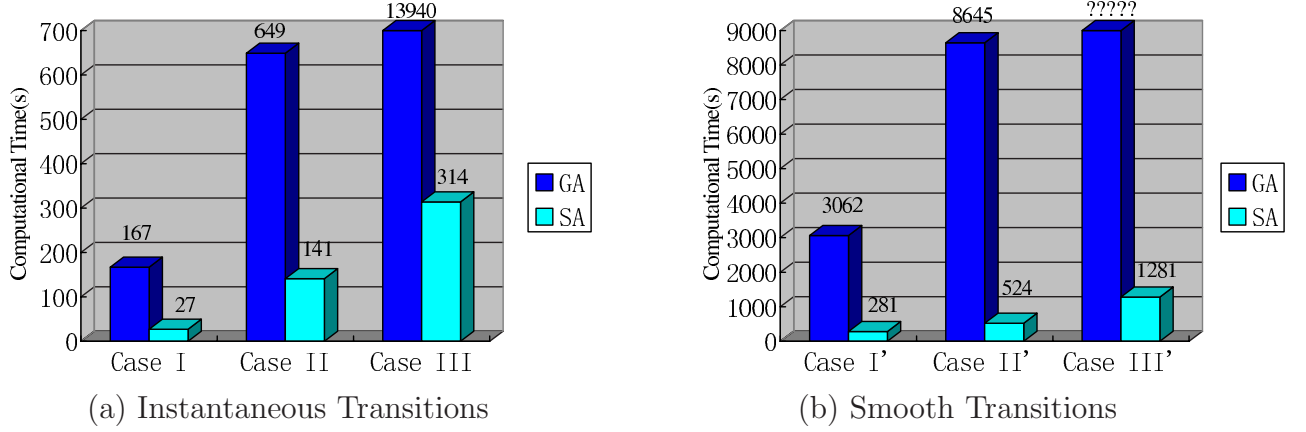


Figure 6.3: Average Computational Time

A comparison of average computation times is shown in Figure 6.3. Since it is almost impossible for the GA to find global optima within the specified upper time bounds, we removed the time constraints to generate this plot. From both Figure 6.3(a) and (b) ⁶, we find SA needs much less time to identify the global optimum; this situation does not change with higher resolution. While the GA does better than the baseline algorithm for high resolution, given the trade-off in probability of attaining the global optimum, the GA does not outperform the baseline when resolution is low.

From both probability of getting to global optima and average computational time measures, SA, specifically ASA, emerges as a fast and accurate method for ter-

⁶Due to the significant computation time, Case III' could not be computed in a given feasible time with the GA.

minal area SNI trajectory optimization. Because our problem is heavily constrained, the GA spends too much time on infeasible solutions, a problem propagated from old to new generations. While SA performance suffers with equality constraints, it performs well with inequality constraints.

6.4 Analysis of Discretization

After identifying ASA as the most efficient alternative algorithm, we studied discretization effects on the final optimal solutions. In this work, we divide the discretization into two groups: the discretization of optimization parameters (γ_i , β_i , and V_i) regulated via *resolution number*; and segment length (distance step) (Δd) regulated via *number of the trajectory segments*.

6.4.1 Effects of Discretization on Design Parameters

As stated above, SA has been identified as an efficient optimization algorithm. Furthermore, its performance does not degrade substantially with resolution increase. This property enables more complex optimizations with higher resolution. Now, we make use of SA for investigating the effects of optimization parameter (γ_i , β_i , and V_i) discretization.

The two categories of test cases are the same as in the previous section, as are initial guess strategy and number of segments ($N = 3$). Solutions are arranged in order of increasing resolution numbers from 8 to 36. We set 500 and 3600 seconds as upper bounds of computational time for instantaneous and smooth transition cases,

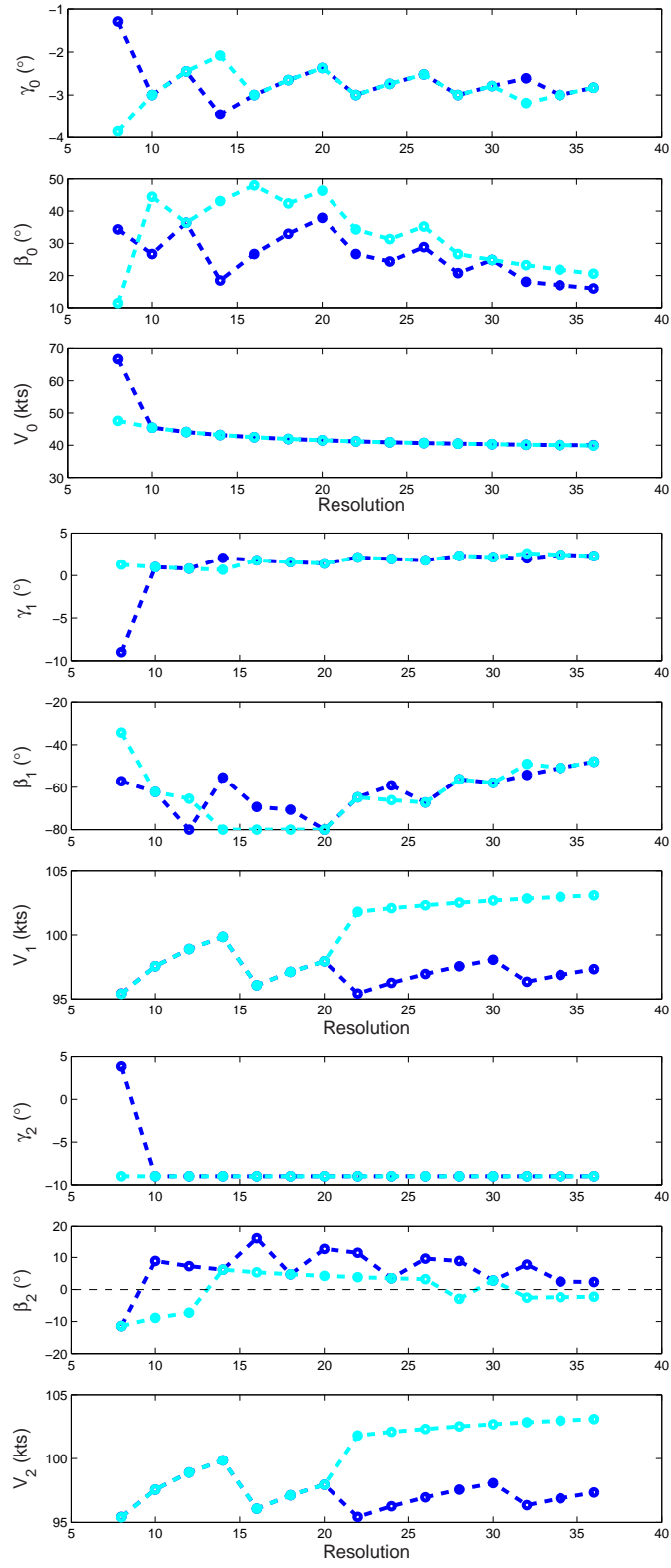


Figure 6.4: Optimal Results for Varied Resolutions

respectively. Similarly, for each case, we compute the global solution and average computational time over 10 realizations. Figure 6.4 shows the optimal values of design variables determined by using SA. The blue lines represent instantaneous transition solutions, while the cyan lines denote solutions with smooth transitions. In most situations the optimal values in these two categories are very close or even the same, which indicates that the inclusion of smooth transitions does not affect the optimization process appreciably except to increase the cost index/ground noise exposure.

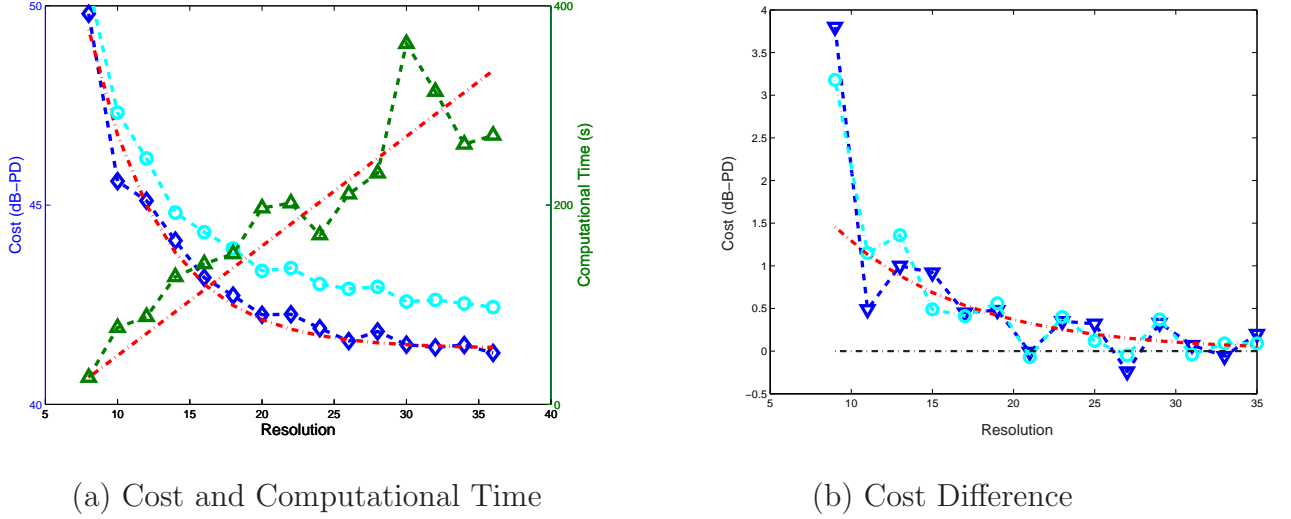


Figure 6.5: Performance with Varying Resolution Number

Figure 6.5(a) shows the best performance indices, with blue lines for instantaneous transition solutions, and cyan lines for smooth transition solutions. Computation time for instantaneous transition solutions is shown (in green) as a function of resolution. Note that in Figure 6.5(a), the globally-optimal cost decreases exponentially with resolution, from 49.80/50.50 dB-PD with resolution 8 to 41.29/42.44 dB-PD with resolution 36. Figure 6.5(b) indicates the difference Δ_{cost} between

neighboring solutions, $\Delta cost = cost(i + 1) - cost(i)$, where i denotes the value of resolution. Both instantaneous and smooth $\Delta cost$ curves approximately decrease along an exponential curve (the red dash-dot line) expressed as: $\Delta cost = e^{-1/8(x-12)}$. With this approximation, we can predict the gap between the minimum cost at resolution 30 and the cost for continuous parameters by calculating $\sum_{i=30}^{\infty} e^{-1/8(x-12)}$, which represents a $\Delta cost$ of less than 1 dB. For continuous design parameters, empirical results indicate SA needs a significant amount of time to reach a globally optimal solution. For example, for an instantaneous transition case, after running 5 hours, 10 realizations indicate the calculated minimum costs are still ~ 44 dB-PD. Thus, it is concluded that discretizing the design variables makes the optimization process more efficient without appreciably increasing cost (less than 1 dB-PD or 3% difference given resolution 32 in the case of 3-segment trajectory optimization). Additionally, from the average trends in Figure 6.5(a), it is also empirically validated that the computation time of SA is proportional to the resolution number for this trajectory optimization problem.

6.4.2 Effects of Segmented Trajectory

We have designed segmented SNI RIA routes [11] to facilitate comprehension by both pilots and ATC. In this section we examine the tradeoff between a low number of segments (simplicity) and trajectory cost. Based on the same test case classes (instantaneous and smooth transitions) with resolution 16, we vary the number of segments between 2 and 5 instead of varying the resolution of the opti-

mization parameters. Table 6.1 shows the initial guesses. Since SA is not highly sensitive to the initial guess, slightly different initial values introduced by different numbers of segments should not introduce a bias. Figure 6.6 presents a comparison of final solution costs and a comparison of computation times. It is shown that the cost differences for both categories (blue for instantaneous transitions and cyan for smooth transitions) decrease at least bi-linearly after the number of segments is greater than 3. As shown in green, the computation time increases rapidly with more than three segment. Thus, for a RIA trajectory from 7 miles out to touchdown, either a 3-segment or 4-segment trajectory provides a good solution given the trade-off between computation time and cost. We expect this result is problem-specific, depending on range and required complexity (to avoid airspace obstacles), but, regardless, from a practical piloting/ATC standpoint, an approach trajectory with more than 4 segments would be difficult to manage without full automation.

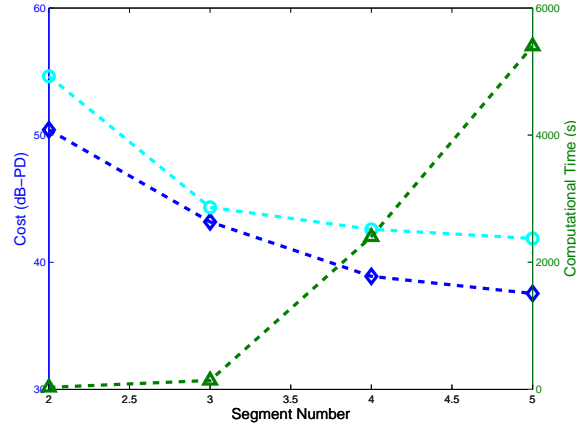


Figure 6.6: Performance with Different Segment Numbers

6.5 Summary

Noise-optimal SNI RIA trajectory optimization requires nonlinear, non-convex, heavily constrained, and discontinuous optimization with no gradient knowledge. In this chapter, we investigated two randomized search methods: the genetic algorithm (GA) and adaptive simulation annealing (ASA). A comparison is presented using an incremental search method as a baseline. It is shown that compared with the baseline, ASA is computationally efficient and expandable to larger problem spaces. ASA has a much higher likelihood of attaining the global optimum than the GA, particularly given a hard computation time constraint. Inclusion of smooth transitions increases the computational time, but provides more realistic cost estimates. Introducing ASA to our problem makes it possible to examine the effects of discretization on the 3D search space. The study shows that, similar to our 2-dimensional analysis, performance increase (cost decrease) is minimal for optimization parameter (γ , β , and V) resolutions above a high value, e.g., 16. Optimization at this level of resolution provides a sufficient near-optimal solution (approximately 3% away from the global optimum with continuous design variables). Finally, an analysis of number of trajectory segments was undertaken to validate the choice of 3-segment solutions given computation time, optimality, and practical operational considerations.

In terms of the analysis in this chapter, if computing speed were increased moderately above the 2.8 GHZ platform used here, the ASA tool could plan at least 3-segment SNI routes in real-time (minutes). Thus, such a tool could be applied to dynamic terminal area trajectory design sensitive both to other traffic, weather, and

noise concerns. This tool could also be a candidate for integration in a high-speed FMS computer, or, at least, sending the pilot/FMS optimal trajectory data from a high speed ATM computing facility on the ground.

It is important to note that our comparison of GA and ASA does not generalize to all optimization problems. For our problem, performance of the GA is diminished by heavy constraints. The GA generally provides multiple, near-optimal solutions, which might be helpful for situations such as dynamic routing planning for multiple aircraft, where initial solution vectors are not obvious. A hybrid algorithm which couples GA and ASA also appears promising for such multi-vehicle trajectory planning problems.

Chapter 7

SNI Trajectory Simulation and Inverse Simulation

To execute the optimal SNI trajectories computed in previous chapters, appropriate commands must be issued to the AH-1 actuators. Either the pilot would issue these commands, or computed commands can be fed into the autopilot. We implemented and analyzed an inverse flight simulation to check the feasibility of the desired SNI trajectories, and to demonstrate the generation of commands associated with desired SNI trajectories.

In this chapter, we first introduce the flight simulation model. Then, the formulation of the optimization-based inverse simulation is presented [60]. Finally, the control outputs of an example SNI trajectory are presented. This sample trajectory is the 3-dimensional noise-optimal trajectory for Sector II with three segments, smooth transitions, and resolution 16.

7.1 Simulation Model

In all trajectory optimization work presented thusfar, we presume a quasi-static point mass model analogous to that modeled by Q-SAM. For this chapter, we utilize a six-degree-of-freedom, quasi-static simulation model for an AH-1 helicopter taken from NASA CR-3144 [59]. In this model, both the lags of the conventional mechanically-actuated hydraulic boost system and the lags of the electronically-

actuated three-axis stability and control augmentation system are neglected. Given a trim state, the linear control system dynamics of the AH-1 are:

$$[s^2C_2 + sC_1 + C_0] \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \end{pmatrix} = [D] \begin{pmatrix} \delta_C \\ \delta_A \\ \delta_B \\ \delta_p \end{pmatrix} + [E] \begin{pmatrix} u_g \\ v_g \\ w_g \\ p_g \\ q_g \\ r_g \end{pmatrix} \quad (7.1)$$

Since we do not model wind effects, we neglect the terms with subscript g , which denotes gust terms. The state variables are thus the velocities x, y, z in an inertial reference frame and Euler angles ϕ, θ, ψ . The controls are typically collective pitch δ_C , longitudinal and lateral cyclic pitches δ_A, δ_B , and rudder pedal δ_p . C_1, C_2, C_0 , and D are matrices, which could be inferred from Table A-1 and A-2 in NASA CR-3144 [59]. All corresponding data are retrieved or linearly-interpolated from the database listed in Table IV-2 and IV-4 in NASA CR-3144 [59].

7.2 Optimization-Based Inverse Simulation

To be applicable, the desired trajectory is reformatted. The entire trajectory is divided into sub-segments⁷ with time intervals of ~ 0.5 seconds. The objective of the inverse simulation is to determine a vector X of design variables that minimize

⁷The desired trajectory has been divided into over 800 sub-segments.

a scalar objective function $F(X)$, subject to constraints.

$$F(X) = \int_0^T [(x - x_D)^2 + (y - y_D)^2 + w_1(z - z_D)^2]^{1/2} dt \quad (7.2)$$

x_D, y_D, z_D represent the desired trajectory. Since values of x_D and y_D are usually much higher than z_D , we set weight $w_1 = 10$. The vector of design variables is composed of four pilot control inputs at preassigned time points during the maneuver:

$$X^T = [\delta_C(t_1)\delta_A(t_1)\delta_B(t_1)\delta_p(t_1), \dots, \delta_C(t_n)\delta_A(t_n)\delta_B(t_n)\delta_p(t_n)] \quad (7.3)$$

where the times t_i , $i = 1, \dots, n$ will be determined by the time separation of the desired trajectory, which corresponds to each sub-segment's time duration (~ 0.5 seconds). The controls are assumed to vary linearly between consecutive time points.

Without automatic stabilization, all helicopters are unstable in hover and in some forward flight configurations. This can affect the trajectory optimization, since the impulse produces relatively large perturbations toward the end of the maneuver and much smaller ones in the first few seconds. The work in [60] shows, due to this instability, if the optimization is over the entire trajectory, the converged solution might not match the required trajectory very well. Thus, the author recommended the optimization be performed over overlapping segments of the trajectory rather than over the entire trajectory ⁸.

In this work, we apply a similar strategy. As in Figure 7.1, each segment

⁸According to the latest development in [62], the author developed a modified algorithm, which can be carried out over a single pass that spans the entire trajectory.

contains six consecutive sub-segments (t_i, \dots, t_{i+5}) with the last 5 $(t_{i+1}, \dots, t_{i+5})$ sub-segments of a given segment overlapping with first 5 of the next. The controls are updated corresponding to t_i, \dots, t_{i+5} , respectively, and only the non-overlapped controls (corresponding to t_i) are stored in the final solution. Since design vector X contains only the controls corresponding to the 6 sub-segments, the total number of design variables is 24. Therefore, the original optimization problem has been replaced by a sequence of smaller sub-problems. By combining the solutions in the first sub-segment of each sub-problem, we obtain the final solution.

The desired trajectory might be composed of accelerating/decelerating segments. Since we assume the velocities could be handled with the throttle or other additional controls, velocity control is neglected in this problem. At the initial time of each sub-problem, the desired velocity is assigned as a part of the trim state. Figure 7.1 shows an example velocity profile. The blue line denotes desired velocities and the red line represents actual velocities.

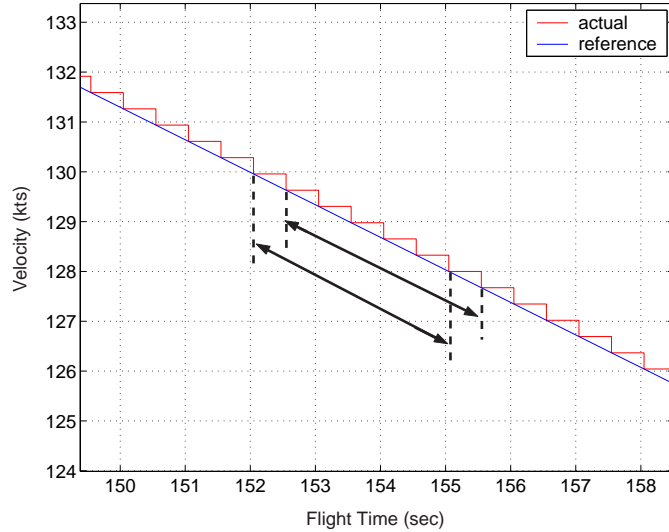


Figure 7.1: Trim Setting of Velocities

7.3 Results

The optimization process is solved by using a standard MATLAB package *fmincon*, which uses the BFGS algorithm [61]. We first set generous constraints⁹ for those four controls, $\delta_C \in [0, 40]deg$, $\delta_A \in [-20, 20]deg$, $\delta_B \in [-28, 28]deg$, and $\delta_p \in [-30, 30]$, respectively. Figure 7.2 shows the resulting actual trajectory and differences from the desired trajectory. In Figure 7.2 (a), the red line denotes the actual trajectory. In this case, the desired trajectory could be followed perfectly with less than 3 ft error. Note that the noise cost is almost identical to the predicted (desired) cost¹⁰.

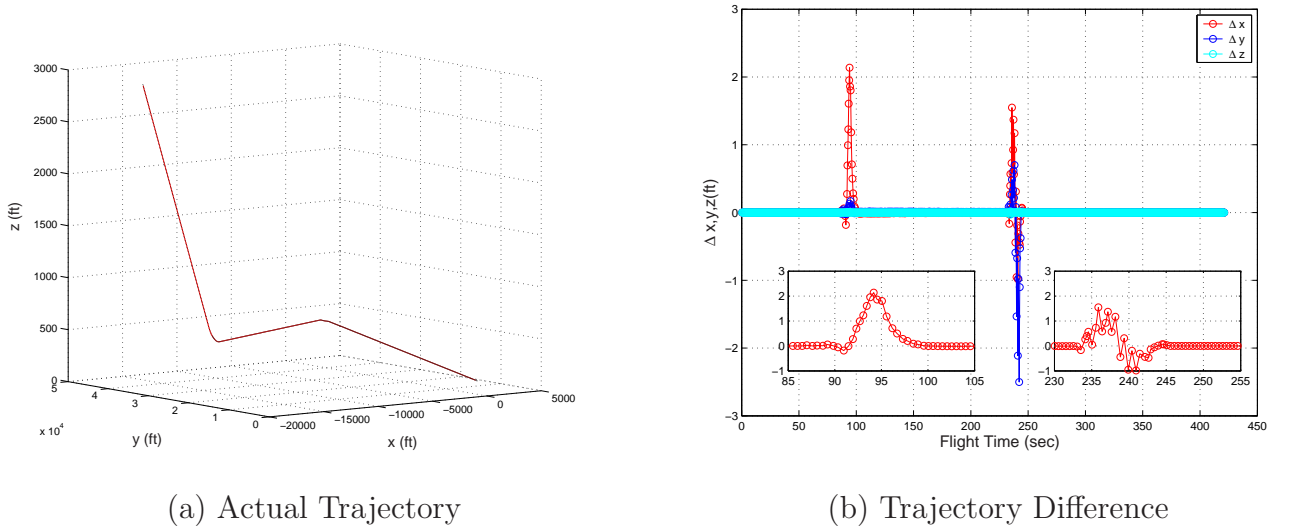
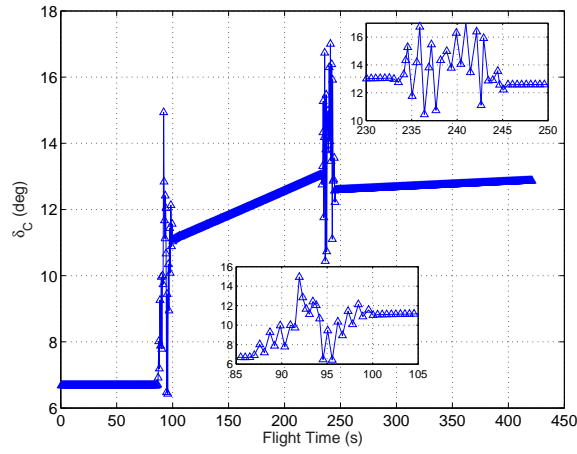


Figure 7.2: Matching Trajectory with Overlapping Segments

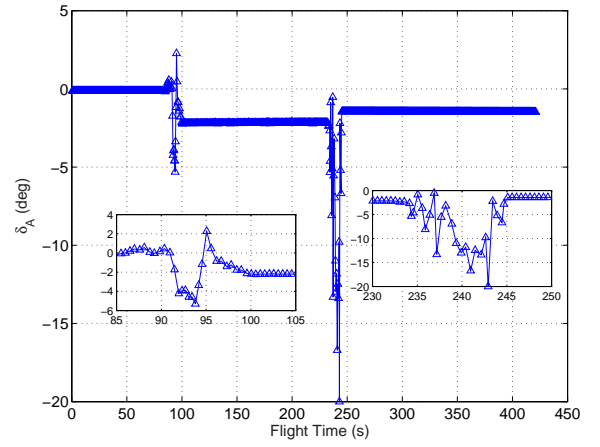
Figure 7.3 presents the corresponding pilot inputs generated by the above optimization process. Here we assume the velocity and acceleration could be followed by adjusting additional controls, as discussed above. In Figure 7.3, it is noted

⁹These nominal settings might not be consistent with manufacturer's.

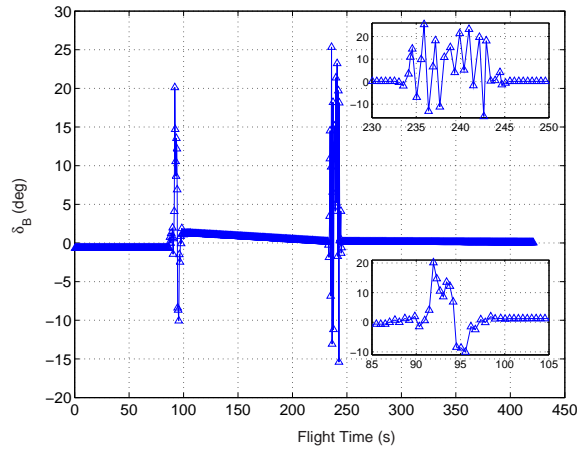
¹⁰As discussed above, the last 5 of over 800 sub-segments were not counted. The cost is rounded to 10^{-2} .



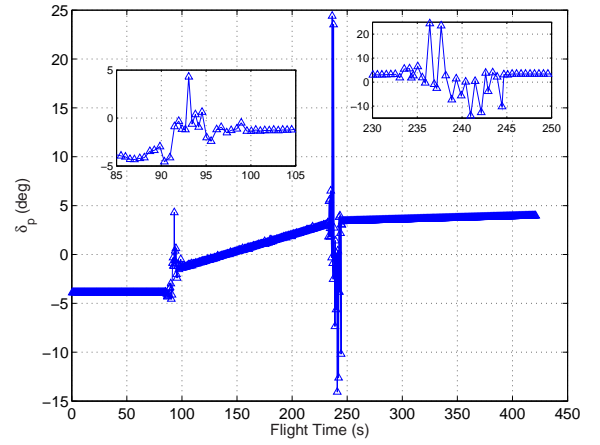
(a) Collective



(b) Lateral Cyclic



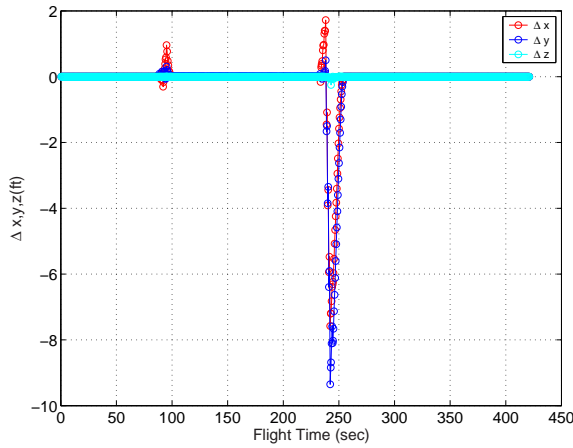
(c) Longitudinal Cyclic



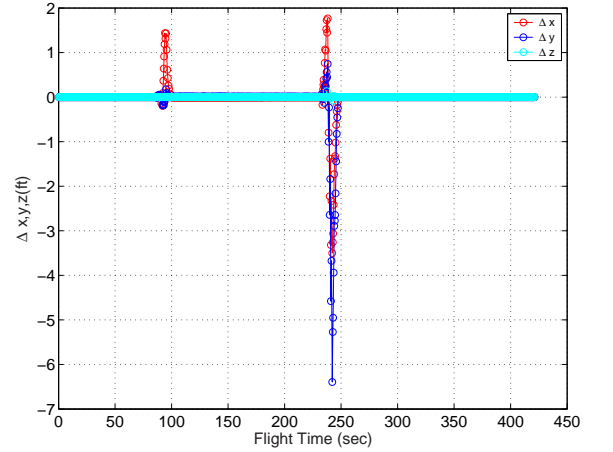
(d) Pedal

Figure 7.3: Time History of Pilot Control Inputs

that some of the control deviations from the trim states are more than 30 degrees (0.52 rads). In terms of the linear model, large control surface deflections make use of a linear model questionable. We introduce constraints to ensure the control variations are sufficiently small. To check the effect of saturations, two cases were executed. In Case I, we set constraints: $\Delta\delta_C \in [-8, 8]deg$, $\Delta\delta_A \in [-6, 6]deg$, $\Delta\delta_B \in [-9, 9]deg$, and $\Delta\delta_p \in [-18, 18]deg$, while in Case II, we set more strict constraints: $\Delta\delta_C \in [-4, 4]deg$, $\Delta\delta_A \in [-3, 3]deg$, $\Delta\delta_B \in [-4.5, 4.5]deg$, and $\Delta\delta_p \in [-9, 9]deg$. Figure 7.4 presents the difference between final and the desired trajectories for Cases I and II. Obviously, constraining the saturation increases the differences, but the trajectories still match very well, since the total costs are still the same. Figure 7.5 provides the pilot controls corresponding to Case II. It is found that the variations of controls are small, which makes the linear model more applicable.



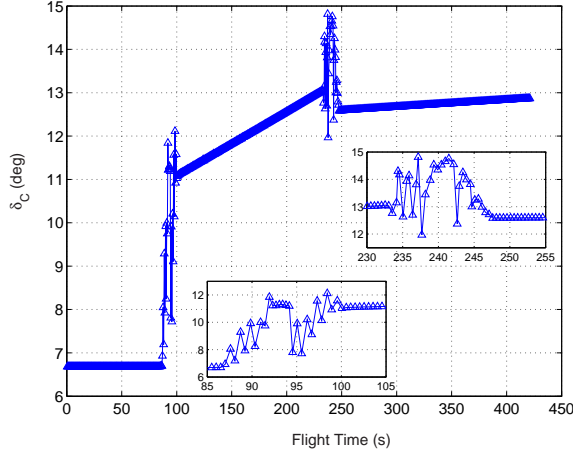
(a) Case I



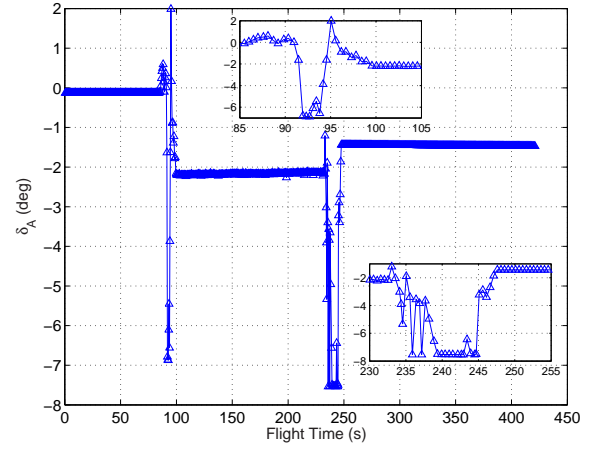
(b) Case II

Figure 7.4: Difference between Actual and Desired Trajectories

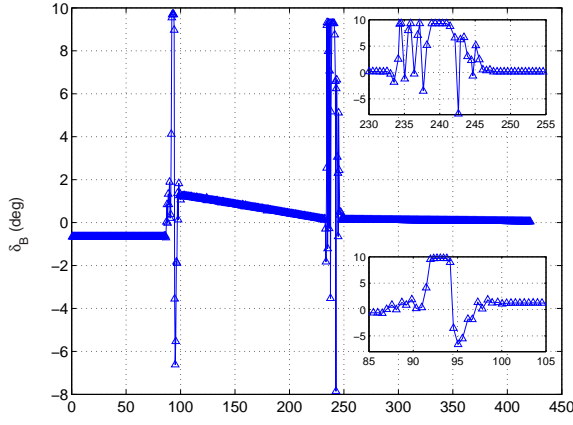
The result of the inverse simulation is a set of pilot/autopilot commands that could be output directly. With these control inputs, the pilot could either follow the



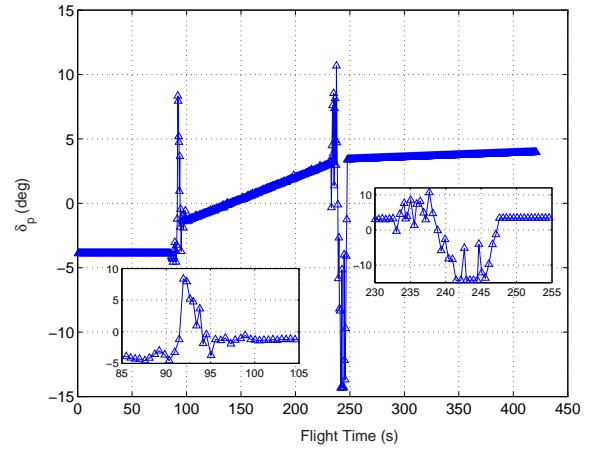
(a) Collective



(b) Lateral Cyclic



(c) Longitudinal Cyclic



(d) Pedal

Figure 7.5: Time History of Pilot Control Inputs

specified time histories or feed them directly into the autopilot. Thus, the whole procedure could be executed automatically, given minor adjustments to compensate for unmodeled perturbations and parameter uncertainties. From the computed commands, we can verify that the desired SNI trajectories are feasible follow by the given control commands. Although, for the entire trajectory, the optimization-based inverse simulation might not be possible in real-time, it could be implemented gradually by calculating the next several segments in advance, with minor real-time

corrections applied to account for wind gusts and other uncertainties.

Chapter 8

Conclusion

This dissertation has studied the design of minimum-noise simultaneous non-interfering (SNI) approach trajectories for Runway Independent Aircraft. As a practical case study, SNI trajectories were designed for an AH-1 rotorcraft on approach to BWI.

This research was divided into three phases. In phase I, we developed a longitudinal plane trajectory planner to study the basic characteristics of a low-noise SNI trajectory. A new *modified cell-decomposition method* was developed. We analyzed BVI noise-optimal trajectories and conducted a sensitivity study over a multi-objective cost function that included noise, time, and fuel. We found there are many local optima in this problem. Inclusion of traffic obstacles or “passenger comfort” constraints forces the optimizer to choose neighboring solutions with near-minimum cost. We also characterized the trade-off between noise-optimal and time/fuel-optimal trajectories, but found that a good balance also exists with appropriate cost function weights.

In phase II, we expanded to 3-D and incorporated models for BWI airport. We modeled population density from a GIS database, and existing fixed-wing flight traffic was encapsulated with impenetrable obstacles. We found viable SNI routes for all approach sectors, and we found that inclusion of population data enables

fewer people to be exposed to high noise levels. To improve the accuracy of our noise estimates, we augmented our segmented trajectory model with smooth flight path angle and turning transitions. It was found that inclusion of smooth transitions increases the overall noise costs but doesn't change the optimal solutions substantially in most cases. A final presentation of DNL maps shows the designed SNI RIA trajectories are acceptable given current community noise standards.

In phase III, given the substantial computation effort required to identify SNI solutions for BWI, we presented work to develop a more efficient real-time implementation through alternative optimization algorithms. After a comparison of GA and SA approaches, we identified adaptive simulated annealing (ASA) as a promising algorithm for this trajectory design problem. Furthermore, based on an analysis of parameter discretization effects, we found that 3 or 4-segment trajectories with resolution of 16 are practical for a future real-time implementation given consideration of both accuracy and time complexity.

Finally, as a follow-up, we also implemented an optimization-based inverse simulation to generate the pilot control commands associated with the specified SNI trajectories. With this tool, both trajectory planning and execution can be fully simulated, validating the SNI trajectories optimized over a more simple but efficient model.

Figure 8.1 summarizes the models and algorithms integrated for this work. There are three types of input data: airport-related (cyan), RIA models (blue), and the noise database (green). When fully integrated, it is anticipated that the pilot/airport planner need only specify landing site and some other preferences,

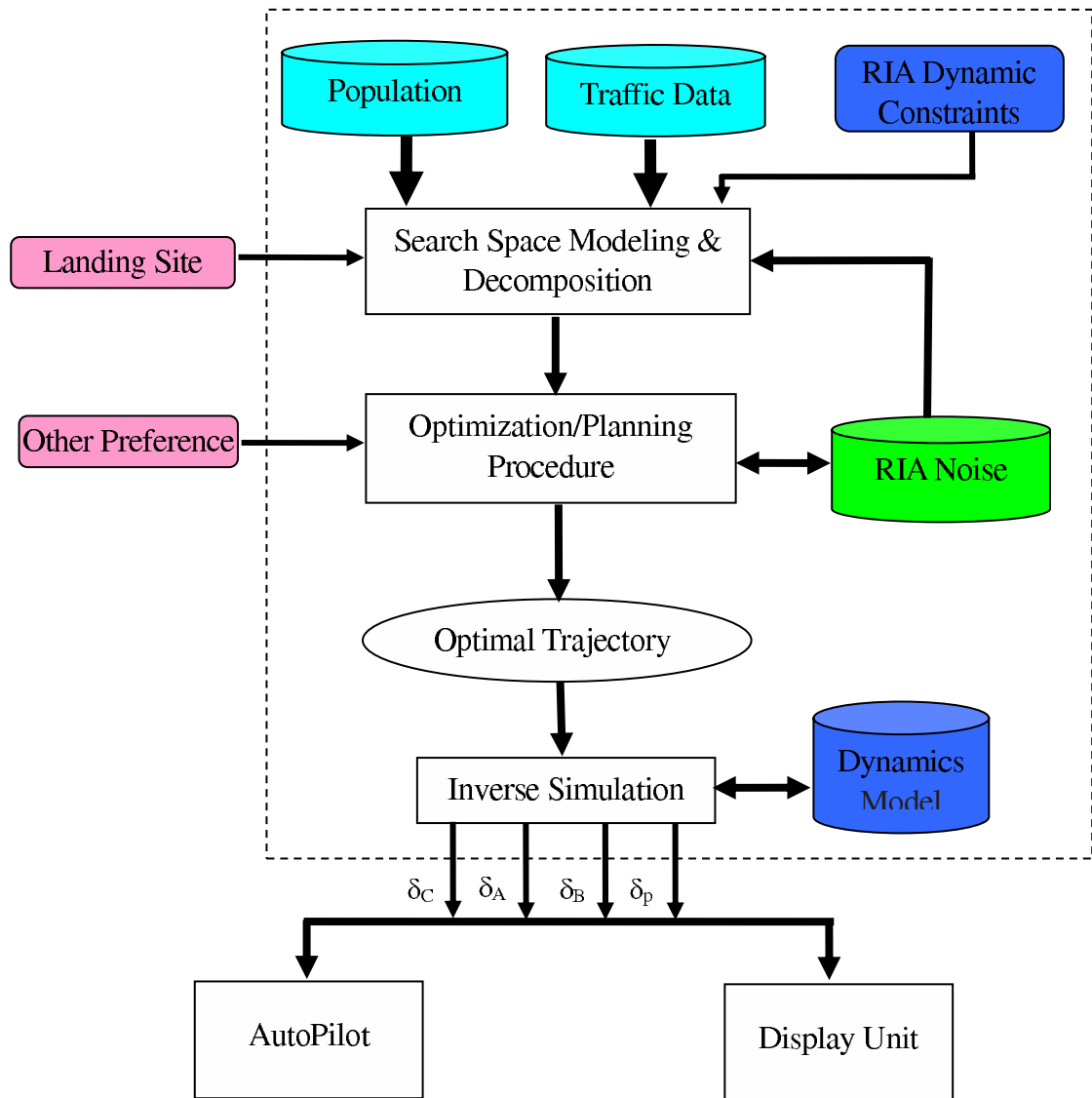


Figure 8.1: Application of Planning Procedure

e.g. cost function weighting factors, then the planner will both compute the SNI trajectory and the control commands necessary to achieve this trajectory.

8.1 Contributions

The contributions of the research in this dissertation are summarized as follows:

- Modification of an existing Approximate Cell-Decomposition algorithm for the 2-dimensional SNI segmented trajectory optimization problem.
- Sensitivity analysis of a multi-objective cost function for 2-D SNI approach trajectories.
- Modeling the 3-dimensional planning problem with an efficient and minimal discrete design parameter set.
- Modeling population density and fixed-wing traffic flows.
- Identification and characterization of optimal SNI trajectories for BWI airport with varied arrival sectors.
- Modeling and analysis of smooth transitions between flight segments in 2-D and 3-D trajectory models.
- Implementing and evaluating genetic algorithm and adaptive simulation annealing optimization approaches for SNI trajectory optimization.
- Analyzing the effects of design parameter discretization.

- Implementing and generating pilot control commands for segmented SNI trajectories via optimization-based inverse simulation.

8.2 Future Work

Runway-independent aircraft are still in the conceptual stage with respect to vehicle class (e.g. rotorcraft, tilt-rotor, ESTOL) and specific vehicle design characteristics. The SNI airspace model and optimization method developed in this work are general for any aircraft, any noise model, and any airport. They can be easily adapted to any types of RIAs, although optimization performance and results will likely be vehicle-dependent. Of course, for each airport, the surrounding population and airspace use must be characterized as modeled in our work, and candidate RIA landing sites must also be identified.

During the entire trajectory planning process, the noise cost was treated as a “black box”. Thus, this trajectory planning procedure could be easily extended to any generic complex costs, e.g. NAPs composed of all noise sources beyond strictly BVI.

Given our work toward a real-time implementation, our planning tools could also be easily integrated into FMS given sufficiently high-speed computing capabilities. With the further integration of inverse simulation, this new FMS module could act as either flight guidance or directly feed the autopilot system.

In general, our work could be applied for any crowded airport area, such as SFO or the New York city area. If SNI solutions exist, our methods could be directly used.

Otherwise, if, for instance, the terminal area airspace is too crowded (e.g. multiple airports in close proximity), the problem could be modified from SNI route design to moving obstacle avoidance, which treats individual flights as dynamic obstacles, instead of modeling the entire corridors as static obstacles. In that case, a hybrid algorithm combining GA and ASA might be promising.

Ultimately, the SNI route planning problem could be extended to handle a mixed fleet of RIA. Of course, cooperation and communication are very important in this scenario, particularly with slow and fast vehicles sharing a landing site. In that case, combining the above suggested methods with market-based coordination [33] might be a promising strategy.

BIBLIOGRAPHY

- [1] J.E. Evans, M. Robinson, and S. Allan, “Quantifying Convective Delay Reduction benefits For Weather/ATM Systems”, *6th USA/Europe Seminar on Air Traffic Management R&D*, Baltimore, MD, June 2005.
- [2] “Boeing Air Traffic Management”, URL:<http://www.boeing.com/atm/flash.html>, [cited Sep. 2005].
- [3] J. Doebbler, P. Gesting, and J. Valasek, “Real-Time Path Planning and Terrain Obstacle Avoidance for General Aviation Aircraft”, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, August, 2005.
- [4] D. Newman, and R. Wilkins, “Rotorcraft Integration into the Next Generation NAS,” *Proceedings of the American Helicopter Society (AHS) 54th Annual Forum*, Washington, DC, May 1998.
- [5] F. H. Schmitz, “Rotor Noise”, Chapter 2 in book authored by H. H. Hubbard, *Aeroacoustics of Flight Vehicles, Theory and Practice, Vol. 1:Noise Sources*, Published for the Acoustical Society of America through the American Institute of Physics, 1995.
- [6] G. Gopalan, M. Xue, E. Atkins, F. H. Schmitz, “Longitudinal-Plane Simultaneous Non-Interfering Approach Trajectory Design for Noise Minimization” *Proceedings of the American Helicopter Society (AHS) 59th Annual Forum*, May, 2003.

- [7] J.-P. Clarke, “Systems Analysis of Noise Abatement Procedures Enabled by Advanced Flight Guidance Technology”, *Journal of Aircraft*, Vol. 37, No. 2, pp. 266-273, March-April 2000.
- [8] K. Elmer, J. Wat, B. Shivashankara, J.-P. Clarke, K. Tong, J. Brown, and A. Warren, “Community Noise Reduction using Continuous Descent Approach: A Demonstration Flight Test at Louisville”, *Proceedings of the 9th AIAA/CEAS Aeroacoustics Conference and Exhibit*, AIAA 2003-3277, Hilton Head, SC, May 2003.
- [9] H. G. Visser, and R. A. A. Wijnen, “Optimization of Noise Abatement Arrival Trajectories”, *Guidance, Navigation, and Control Conference and Exhibit*, AIAA 2001-4222, Montreal, Canada, Aug. 6-9, 2001.
- [10] H. G. Visser, and R. A. A. Wijnen, “Optimization of Noise Abatement Departure Trajectories”, *Journal of Aircraft*, Vol. 38, No. 4, pp.620-627, July-August 2001.
- [11] F. J. Vormer, M. Mulder, M. M. van Paassen, and J. A. Mulder, “Design and Preliminary Evaluation of Segmented-based Routing Methodology”, *Proceedings of Guidance, Navigation, and Control Conference*, AIAA, Monterey, CA, August 2002.
- [12] P. Hagelauer and F. Mora-Camino, “A Soft Dynamic Programming Approach for On-line Aircraft 4d-trajectory Optimization”, *European Journal of Operational Research*, Vol. 107, pp. 87-95, 1998.

- [13] M. Pachter and J. Hebert, “Optimal Aircraft Trajectories for Radar Exposure Minimization”, *Proceedings of American Control Conference*, June 2001.
- [14] J. E. Bobrow, “Optimal Robot Path Planning Using the Minimum-time Criterion”, *IEEE Journal of Robotics and Automation*, Vol.4, No.4, 1988.
- [15] H. Tominaga and B. Bavarian, “Global Robot Path Planning Using Exact Variational Methods”, *Proceedings of IEEE International Conference on System, Man, and Cybernetics*, pp. 617-619, 1990.
- [16] S. Sunder and Z. Shiller, “Optimal Obstacle Avoidance Based on the Hamilton-jacobi-bellman Equation”, *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 2, April, 1997.
- [17] J. Betts, “Survey of Numerical Methods for Trajectory Optimization”, *Journal of Guidance, Control, and Dynamics*, Vol. 21, pp. 193-207, 1998.
- [18] P. F. Gatch, “CAMTOS - A Software Suite Combining Direct and Indirect Trajectory Optimization Methods”, Dissertation, Institut für Flugmechanik und Flugregelung, Universität Stuttgart, 2002.
- [19] J. C. Latombe, “Robot Motion Planning”, Kluwer Academic Press, 1991.
- [20] A. M. Ladd and L. E. Kavraki, “Measure Theoretic Analysis of Probabilistic Path Planning”, *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 2, pp. 229-242, 2004.

- [21] J. J. Kuffner and S. M. LaValle, “RRT-connect: An efficient approach to single-query path planning”, In *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 995–1001, 2000.
- [22] F. Lamiriaux, and L. E. Kavraki, “Path Planning for Elastic Plates Under Manipulation Constraints”. In *Proceedings of The IEEE International Conference on Robotics and Automation (ICRA)*, pp. 151-156, IEEE Press, Detroit, MI, May 1999.
- [23] R. H. Wilson, L. E. Kavraki, J.-C. Latombe, and T. Lozano-Perez, “Two-Handed Assembly Sequencing”. *International Journal of Robotics Research*, Vol. 14, No. 4, 335-350, 1995.
- [24] K. Hauser, T. Bretl, J.C. Latombe, “Learning-Assisted Multi-Step Planning”, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005
- [25] R. Tombropoulos, J.C. Latombe, and J.R. Adler. “A General Algorithm for Beam Selection in Radiosurgery”, *IARP Workshop on Medical Robotics*, Vienna, pp. 91-98, 1996.
- [26] R.Z. Tombropoulos, J.R. Adler, and J.C. Latombe, “CARABEAMER: A Treatment Planner for a Robotic Radiosurgical System with General Kinematics”, *Medical Image Analysis*, Vol. 3, No. 3, pp.237-264, 1999.
- [27] J. Kuffner and J.C. Latombe, “Interactive Manipulation Planning for Animated Characters”, *Proceeding of Pacific Graphics*, Hong Kong, October, 2000.

- [28] P. Cheng, Z. Shen, and S. M. LaValle, “RRT-based Trajectory Design for Autonomous Automobiles and Spacecraft”, *Archives of Control Science*, Vol. 11, No. 3-4, pp. 167-194, 2001.
- [29] R. Beard, T. McLain, M. Goodrich, and E. Anderson, “Coordinated Target Assignment and Intercept for Unmanned Air Vehicles”, *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 6, December, 2002.
- [30] R. Bellman, “Dynamic Programming”, *Princeton University Press*, 1957.
- [31] P. Hagelauer, “Contribution a l’Optimisation Dynamique de Trajectoires de Vol pour un Avion de Transport”, Ph.D, Dissertation, CNRS - Universite Paul Sabater de Toulouse, France, June 1997.
- [32] C. Tomlin, I. Mitchell, and R. Ghosh, “Safety Verification of Conflict Resolution Maneuvers”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 2, No. 2, June, 2001.
- [33] A. Pongpunwattana, R. Rysdyk, “Real-Time Planning for Multiple Autonomous Vehicles in Dynamics Uncertain Environments”, *Journal of Aerospace Computing, Information, and Communication*, Vol. 1, pp. 580-604, December, 2005.
- [34] R. A. Brooks, and T. Lozano-Perez, “A Subdivision Algorithm in Configuration Space for Findpath with Rotation”, *Proceedings of the 8th International Conference on Artificial Intelligence*, pp. 799-806, 1983.

- [35] D. P. Bertsekas, “Dynamic Programming and Optimal Control”, *Athena Scientific*, 1995.
- [36] E. George, “Optimization of Satellite Constellations for Discontinuous Global Coverage via Genetic Algorithm”, *AAS/AIAA Astrodynamics Specialist Conference*, Sun Valley, Idaho, 1997.
- [37] S. Russell, and P. Norvig, “Artificial Intelligence: a Modern Approach”, *Prentice Hall Series*, New Jersey, 1995.
- [38] T. Dean, L. P. Kaelbling, and A. Nicholson, “Planning under Time Constraints in Stochastic Domains”, *Artificial Intelligence*, Vol. 76, No. 1-2, pp. 35-74, 1995.
- [39] J. Albus, N. DeClaris, A. Lacaze, and A. Meystel, “Neural Network Based Planner/Learner for Control Systems”, *Proceedings of the 1997 International Conference on Intelligent Systems and Semiotics*, Gaithersburg, MD, 1997.
- [40] W. A. Crossley and E. A. Williams, “Simulated Annealing and Genetic Algorithm Approaches for Discontinuous Coverage Satellite Constellation Design”, *Engineering Optimization*, Vol. 32, pp. 353-371, 2000.
- [41] W. Johnson, “Helicopter Theory”, *Dover Publications*, New York, 1994.
- [42] G. Gopalan, F. H. Schmitz, and B. W. Sim, “Flight Path Management and Control Methodology to Reduce Helicopter Blade-Vortex (BVI) Noise” *Proceedings of the American Helicopter Society (AHS) Vertical Lift Aircraft Design Conference*, January, 2000.

- [43] F. H. Schmitz, “Reduction of Blade-Vortex Interaction (BVI) Noise Through X-Force Control”, NASA TM-110371, April 1995.
- [44] E. M. Atkins, and M. Xue, “Noise-Sensitive Final Approach Trajectory Optimization for Runway-Independent Aircraft”, *Journal of Aerospace Computation, Information and Communication*, Vol.1 pp. 269-287, July, 2004.
- [45] “Quarterly Noise Report - Fourth Quarter 2002”, Airport Noise and Abatement Office, Maryland Aviation Administration.
- [46] “Census 2000 TIGER/Line Data”, URL:http://www.esri.com/data/download/census2000_tigerline/index.html, [cited Aug. 2004].
- [47] M. Xue, and E. M. Atkins, “Noise-Minimum Runway-Independent Aircraft Approach Design for Baltimore-Washington International Airport”, *Journal of Aircraft*, To Appear.
- [48] J. J. Craig, “Introduction to Robotics: Mechanics and Control, Second Edition”, *Addison-Wesley Publishing Company*, 1989.
- [49] D. A. Boxwell, F. H. Schmitz, W. R. Splettstoesser, and K. J. Schultz, “Helicopter Model Rotor-Blade Vortex Interaction Impulsive Noise: Scalability and Parametric Variations”, *Journal of American Helicopter Society*, Vol. 32, No. 1, 1987.
- [50] B. W. Sim, and F. H. Schmitz, “Blade-Vortex Interaction (BVI) Noise: Retreating Side Characteristics, Sensitivity to Chordwise Loading and Unsteady

- Aerodynamics”, *American Helicopter Society (AHS) Aeromechanics Specialists’ Meeting*, Atlanta, Georgia, 2000.
- [51] Z. Michaelwicz, “Genetic Algorithms + Data Structures = Evolution Programs, Third Edition”, *Springer-Verlag Berlin Heidelberg*, 1996.
 - [52] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller, “Equation of State Calculations by Fast Computing Machines”, *The Journal of Chemical Physics*, Vol. 21, No. 6, pp. 1087-1092, 1953.
 - [53] S. Y. Geman, and D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, pp. 721-741, 1984.
 - [54] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing”, *Science*, Vol. 220. No. 4598, pp. 671-680, 1983.
 - [55] H. Szu, and R. Hartley, “Fast Simulated Annealing”, *Physics Letters A*, Vol. 122, No. 3-4, pp. 157-162, 1987.
 - [56] B. Hajek, “Cooling Schedule for Optimal Annealing”, *Mathematics of Operations Research*, Vol. 13, No. 2, pp. 311-329, 1988.
 - [57] L. Ingber, “Very Fast Simulated Re-Annealing”, *Mathl. Comput. Modelling*, Vol. 12, pp. 967-973, 1989.
 - [58] L. Ingber, “Simulated Annealing: Practice versus theory”, *Mathl. Comput. Modelling*, Vol. 18, No. 11, pp. 29-57, 1993.

- [59] R. K. Heffley, W. F. Jewell, J. M. Lehman, and R. A. Vanwinkle, “A Compilation and Analysis of Helicopter Handling Qualities Data. Volume 1: Data Compilation”, NASA CR 3144, August, 1979.
- [60] R. Celi, “Optimization-Based Inverse Simulation of a Helicopter Slalom Maneuver”, *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 2, 2000.
- [61] R. Fletcher, “A New Approach to Variable-Metric Algorithms”, *Computer Journal*, Vol. 13, pp. 217-322, 1970.
- [62] R. Celi, “Simulation and Analysis of ADS-33 Quickness Maneuvers for An Articulated Rotor Helicopter”, *Proceedings of the American Helicopter Society (AHS) 61th Annual Forum*, June, 2005.